Departamento de Informática

PROYECTO FIN DE CARRERA
Ingeniería Técnica en Informática de Gestión

MINERÍA DE SENTIMIENTOS SOBRE TWITTER

Autor: ANTONIO MARTÍNEZ RODRÍGUEZ

Tutora: ANABEL FRAGA VÁZQUEZ

Leganés, Octubre de 2015

Título: MINERÍA DE SENTIMIENTOS SOBRE TWITTER

Autor: ANTONIO MARTÍNEZ RODRÍGUEZ

Directora: ANABEL FRAGA VÁZQUEZ

EL TRIBUNAL

Presidente: _____

Vocal: _____

Secretario: _____

Realizado el acto de defensa y lectura del Proyecto Fin de Carrera el día __ de Octubre de 2015 en Leganés, en la Escuela Politécnica Superior de la Universidad Carlos III de Madrid, acuerda otorgarle la CALIFICACIÓN de

VOCAL

SECRETARIO PRESIDENTE

Agradecimientos

Este Proyecto Fin de Carrera supone el cierre de mi titulación en Ingeniería Técnica en Informática de Gestión. Titulación y paso por la universidad que me ha abierto el camino hacia lo que hoy soy tanto a nivel profesional como en gran medida a nivel personal.

Llegados a este momento no puedo por menos que acordarme y agradecer desde lo más profundo de mi corazón a todas las personas que me han rodeado durante estos años y que tengo la fortuna de que sigan estando a mi lado.

En primer lugar y de forma muy especial a mis padres, Antonio y Mari, sin cuyo apoyo, cariño y confianza ahora no estaría escribiendo estas palabras.

A mi mujer, Elena, todavía recuerdo el momento en el que el azar cruzó nuestros caminos en la asignatura de Bases de Datos Avanzadas, nunca una asignatura ha significado tanto para mí.

A mis hijos Alejandro e Irene, por entender las largas jornadas de trabajo de papá y estar siempre ahí, haciéndome tener sentimientos imposibles de explicar.

No quiero olvidar a todas aquellas personas de la generación de mis abuelos que lucharon para que lo que entonces era el privilegio de unos cuantos, hoy sea la fortuna de todos: poder acceder a la educación con independencia de tu condición.

Para terminar me gustaría agradecer a mi tutora, Anabel Fraga Vázquez, el que haya aceptado tutelar el presente trabajo, espero que el resultado sea de su agrado.

Resumen

El desarrollo actual de Internet así como su nivel de expansión y accesibilidad lo ha convertido en un medio ideal sobre el que construir herramientas para la comunicación entre personas.

Las redes sociales han logrado introducirse en el día a día de nuestra sociedad, convirtiéndose en esas herramientas que explotan las características comunicativas de Internet a la hora de intercambiar información y compartir intereses e inquietudes.

Sobre este entorno ha crecido la necesidad de extraer conocimiento, lo que unido al creciente interés por el desarrollo y conocimiento de la inteligencia emocional, da origen a trabajos como el que se realiza en este proyecto.

En el proyecto se toman los datos de un perfil de Twitter creado en la asignatura Ingeniería de Sistemas de Información del Master Universitario en Ingeniería Informática. Perfil dirigido a la interacción con los alumnos para conocer las opiniones, inquietudes e intereses que se despiertan sobre las diferentes materias, clases, presentaciones y otros trabajos realizados en la asignatura.

El proyecto establece un marco de trabajo para extraer conocimiento de los *tweets* publicados.

- En primer lugar extrayendo los *tweets* de Twitter.
- A continuación transformándolos a lenguaje natural.
- En tercer lugar realizando una minería de sentimientos apoyada en el desarrollo de una ontología.
- Indexando los *tweets* sobre la ontología.
- Finalmente estudiando el resultado de la indexación y transformándolos en conocimiento.

Abstract

Nowadays Internet has become in a vehicle for people to communicate in many ways because of its current development, expansion and high accessibility.

Moreover the inclusion of social networks in the day to day of our society, have made them an ideal place to exchange information and share interests and concerns.

This project is based on this environment and the need to extract knowledge and in particular, emotional knowledge.

The project draws on *tweets* from a Twitter profile created in the Information Systems Engineering course of Informatics Engineering Master. The profile is aimed to interact with the students to discover opinions, concerns and other interests caused by the classes, presentations and other class works.

The project established a framework to extract knowledge about the published *tweets.*

- First, extracting the *tweets* from Twitter.
- Next, transforming the *tweets* into a natural language statements.
- Third, developing an ontology to represent the *tweets* universe which is the base for the sentiment mining.
- Then, defining and indexing *tweets* into the ontology.
- And finally, analyzing the results and transforming them on knowledge.

Índice general

Índice de figuras

Índice de tablas

Capítulo 1

Introducción y Objetivos

Este primer capítulo introduce el Proyecto de Carrera detallando los objetivos del mismo así como la estructura y desarrollo de la memoria.

1. Introducción y Objetivos

1.1 Introducción

En Proyecto Fin de Carrera se procede con la ejecución de técnicas de minería de datos, en particular minería de sentimientos sobre las aportaciones de los alumnos a través de la red social Twitter en la asignatura Ingeniería de Sistemas de Información del Master Universitario en Ingeniería Informática.

Se pretende obtener conclusiones a partir del análisis de dichos datos de forma que se detecten las opiniones, inquietudes e intereses que despierten las diferentes sesiones, materias y trabajos realizados en la asignatura.

1.2 Objetivos

El Objetivo principal consiste en construir un modelo de conocimiento que sirva como base para el proceso de mejora continua en la labor docente de la asignatura.

Podemos también enumerar un conjunto de objetivos secundarios y que sirven para la consecución del objetivo principal.

- Extracción de información de Twitter

  En el proyecto se tiene que analizar e implementar un mecanismo de extracción de la información que se publique en el perfil de la asignatura (miisi_uc3m).

- Transformación/Preparación

  También es objetivo del proyecto recoger los datos anteriores y transformarlos a un formato que permita que puedan ser procesados informáticamente.

- Modelo de conocimiento

  El procesamiento ha de ser posible por herramientas informáticas que faciliten tanto su procesamiento actual como su escalado y reutilización.

1.3 Organización de la Memoria

La memoria consta de los siguientes apartados.

- Estado de la Cuestión

  Se inicia el análisis con una revisión del estado de la cuestión.
  En primer lugar se realiza una retrospectiva a los orígenes de internet para entender el surgir de las redes sociales.

Se continua con un repaso de los hechos tecnológicos que han concurrido en la creación de las disciplinas de la minería de datos y en concreto la minería de sentimientos.

Finaliza este primer análisis con la introducción del concepto de minería de sentimientos así como de las diferentes tecnologías y técnicas que lo rodean.

- Análisis y Diseño

  Entendido el problema y puestos en contexto tecnológico e histórico, en este apartado se desarrolla en profundidad el análisis y diseño de las diferentes fases de la implementación del proyecto.

  También se detallará en este apartado la metodología utilizada así como el entorno tecnológico completo necesario para la implementación.

- Planificación y presupuesto

  Se presenta el plan de proyecto con las tareas y recursos necesarios para la ejecución del presente trabajo.

  Así mismo se hace una valoración económica correspondiente a los costes incurridos tanto en personal como en recursos materiales o licencias software.

- Extracción de la Información

  Este capítulo inicia el detalle del proceso de implementación.

  El proceso de minería se inicia con la extracción de datos de la fuente Twitter y su tratamiento para cargarlo en el sistema.
  En este apartado se describe en detalle el proceso de extracción y transformación de la información para posibilitar su modelado y posterior análisis.

  Se analiza el desarrollo de una utilidad dentro del proyecto para proceder con la transformación de *tweets*.

- Minería de sentimientos

  El apartado desarrolla las técnicas de análisis y procesamiento de la información utilizados.

  Se detalla la creación de una ontología sobre la herramienta knowledeMANAGER.

  Se describe la creación de los términos, reglas de tratamiento de tokens así como la creación de un modelo de patrones que definen la sintaxis a la que se ajustan los *tweets*.

  Se completa el modelo de conocimiento mediante las relaciones entre patrones, definición de meta-propiedades y finalmente se ejecuta un proceso de indexación de todos los *tweets* recogidos sobre el modelo diseñado.

- Conclusiones.

  Sobre el resultado del proceso de indexación se muestran datos estadísticos con el resultado de los análisis efectuados.

  La memoria se completa con un apartado dedicado a enumerar otros trabajos que no siendo necesarios para el presente proyecto sí se han detectado durante la ejecución del mismo y pueden ser de utilidad y ayudar a ampliar el trabajo del proyecto.

Capítulo 2

Estado de la Cuestión

El primer paso en la ejecución del proyecto consiste en analizar la problemática y el estado de las tecnologías y metodologías que se van a utilizar para la consecución de los objetivos marcados.

2. Estado de la Cuestión

En este apartado se desarrolla una introducción a las técnicas de minería de datos y minería de sentimientos en las que se basa este Proyecto Fin de Carrera.

En primer lugar se realiza un recorrido histórico desde las primeras páginas interactivas hasta llegar al fenómeno de las redes sociales.

Por otro lado, también se introduce el proceso de descubrimiento de conocimiento basado en la minería de datos y en particular de la minería de sentimientos.

2.1 Antecedentes Históricos Redes Sociales

El Conocimiento es una facultad propia del ser humano, que nos diferencia de otras especies y que durante toda nuestra Historia ha sido sinónimo de desarrollo.

El Conocimiento se adquiere mediante la educación, la experimentación, la relación con el entorno y el estudio y asimilación de la información que nos rodea.

Sin entrar a realizar un estudio del Conocimiento Humano, resulta evidente que en los últimos años y con el desarrollo tecnológico actual, está a nuestra disposición una cantidad inagotable de Información sobre cualquier cuestión que resulte de nuestro interés. Uno de los objetivos fundamentales de la minería de datos es precisamente la transformación de la información en conocimiento.

En este sentido el desarrollo de Internet y su masiva actual accesibilidad tienen un papel fundamental. No en vano las propias Naciones Unidas han declarado la facilidad de acceso a Internet como un derecho fundamental ya que promueve el progreso de la sociedad en su conjunto [1].

Ya en los años 60 los pioneros de Internet vaticinaban la necesidad de una red de comunicación global que actuase a modo de biblioteca universal y que fuese capaz de conectar múltiples computadoras y proporcionar la simbiosis entre el hombre y la máquina [2].

Desde el punto de vista tecnológico y de comunicaciones, los siguientes datos no hacen más que mostrar la evidencia del desarrollo de Internet.

Desde los primeros años marcados con el origen militar y la conexión entre las universidades de UCLA, el Stanford Research Institute, la Universidad de Utah, la Universidad de California, Santa Bárbara y el posterior acceso europeo desde Londres y Noruega.

Año	Nodos Conectados
1969	Primeros dos nodos en ARPANET.
1973	Primeras conexiones desde Europa a ARPANET.
1984	1.000 computadoras conectadas.
1987	10.000 computadoras conectadas.
1989	100.000 computadoras conectadas.

Año	Nodos Conectados
1992	1 millón de computadoras conectadas.

Tabla 1 - Primeros Nodos conectados a Internet

Hasta el momento actual en el que el número de Hosts conectados se cuenta por miles de millones [3].

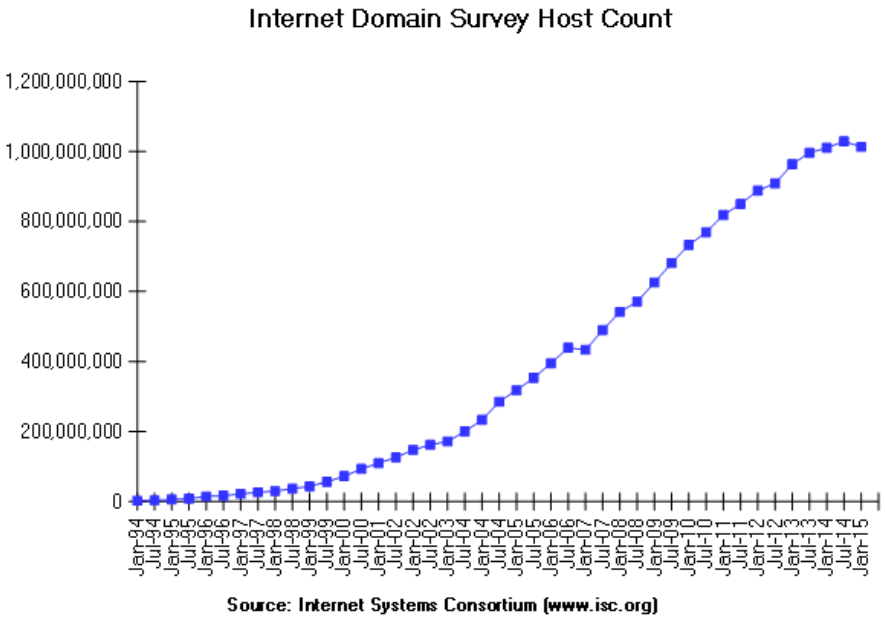

Ilustración 1 – Nodo conectados a Internet

Esto no es más que la consecuencia de que el número de usuarios también se cuente por miles de millones [4].

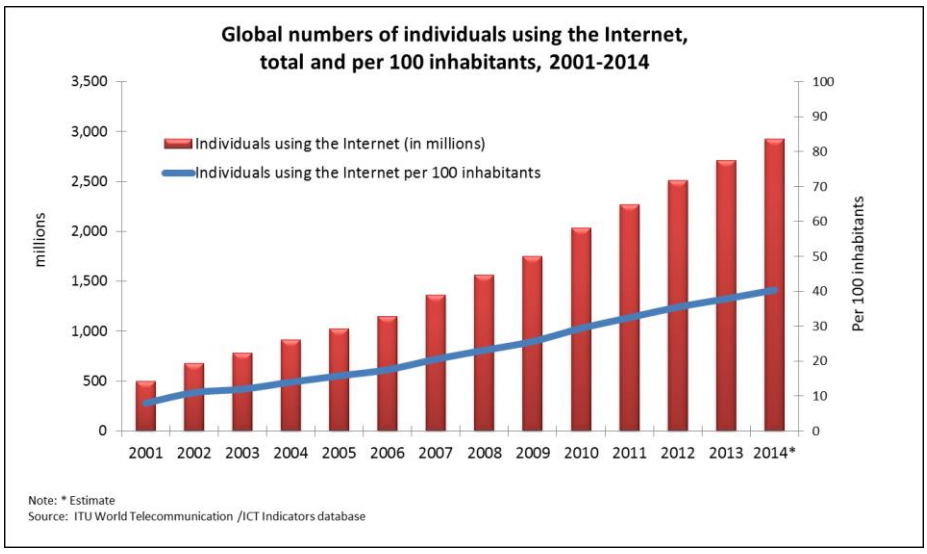

Ilustración 2 – Evolución usuarios de Internet

Sin duda alguna este crecimiento está relacionado con los servicios prestados sobre la red. Es a finales de los 80 y principios de los años 90, con la creación de HTML/HTTP en el CERN por Tim Berners-Lee [5], el desarrollo de los primeros navegadores con interfaz gráfico y la incorporación de ISPs a la red, cuando se

incorporan servicios privados orientados a usuarios personales que a su vez también se van incorporando según se van popularizando los PC y navegadores como Mosaic o Netscape.

2.1.1 Redes Sociales

Podemos entender una Red Social como el servicio prestado en una plataforma de Internet en el que a partir de un perfil se puede mantener una *bitácora* personal con compartición de contenidos con otros usuarios de la red formando de esta forma las conexiones sociales que definen a la propia red.

Típicamente la Red Social ofrece al usuario herramientas para localizar otros usuarios en base a propiedades del propio usuario como los colegios, institutos y/o universidades en los que ha estudiado, perfil e inquietudes profesionales, compañías para las que ha trabajado, aficiones, etc. Facilitando de esta forma el crecimiento de los enlaces entre diferentes usuarios de la red y en consecuencia de la propia red.

Desde los comienzos de Internet han existido servicios basados en comunidades interesadas en las mismas cuestiones.

A finales de los años 70 ya se empezaban a utilizar los grupos de discusión en la red USENET que aunque estaban muy claramente protagonizados por cuestiones relacionadas con la propia tecnología y el desarrollo de Internet, también empezaron a surgir grupos relacionados con aficiones personales como la ciencia ficción o el cine.

Es con la popularización de Internet con la que se empiezan a crear los primeros *blogs*, estamos hablando ya de mediados de los 90 cuando los *blogs* no eran más que actualizaciones de páginas personales a modo Bitácora.

La posterior creación de herramientas como blogger (1999 [6]) o wordpress (2003 [7]) no hace sino facilitar la creación de *blogs* y de esta forma popularizar y convertir los *blogs* en un fenómeno de masas con un alto componente social.

Algunas publicaciones denominan a este tipo de servicios Redes Sociales Indirectas.

"*Son redes sociales indirectas aquellas cuyos servicios prestados a través de Internet cuentan con usuarios que no suelen disponer de un perfil visible para todos existiendo un individuo o grupo que controla y dirige la información o las discusiones en torno a un tema concreto. Resulta especialmente relevante aclarar que este tipo concreto de redes sociales son las precursoras de las más recientes redes sociales directas desarrolladas dentro del nuevo marco de la Red 2.04. Las redes sociales indirectas se pueden clasificar en foros y blogs*" [8]

Finalmente es a partir de este punto en el que se lanzan diferentes plataformas que de una forma u otra dan un mayor protagonismo al usuario.
Con los *blogs* no deja de ser un usuario o pequeño grupo de usuarios los que gestionan y moderan la bitácora. Si bien es cierto que con las herramientas anteriormente nombradas la creación y mantenimiento de un *blogs* no implica altos conocimientos técnicos, sí puede resultar un elemento de rechazo en usuarios poco familiarizados con la tecnología.

La introducción de plataformas como Facebook, Twitter, Tumblr, MySpace, Instagram, LinkedIn, Flickr, etc. sigue un enfoque diferente. Se proporciona una plataforma común en la que un usuario, mediante la creación de un perfil, gestiona su propia *bitácora*. Adicionalmente, de forma muy sencilla se proporcionan enlaces entre diferentes perfiles creando de esta forma interconexiones entre usuarios, es decir, un conjunto de Redes Sociales. No exigiendo en ningún caso conocimientos técnicos sobre Informática o diseño Web.

Alcanzamos con este tipo de servicios lo que podemos calificar como Redes Sociales directas.

"*Son redes sociales directas aquellas cuyos servicios prestados a través de Internet en los que existe una colaboración entre grupos de personas que comparten intereses en común y que, interactuando entre sí en igualdad de condiciones, pueden controlar la información que comparten. Los usuarios de este tipo de redes sociales crean perfiles a través de los cuales gestionan su información personal y la relación con otros usuarios. El acceso a la información contenida en los perfiles suele estar condicionada por el grado de privacidad que dichos usuarios establezcan para los mismos*" [8].

Las Redes Sociales son un fenómeno que desde el lanzamiento en 1995 de *classmates.com* [9] han proliferado de forma que en la actualidad ocupan una gran parte del tráfico de Internet como puede apreciarse en las siguientes gráficas del portal *Alexa.com* [10]

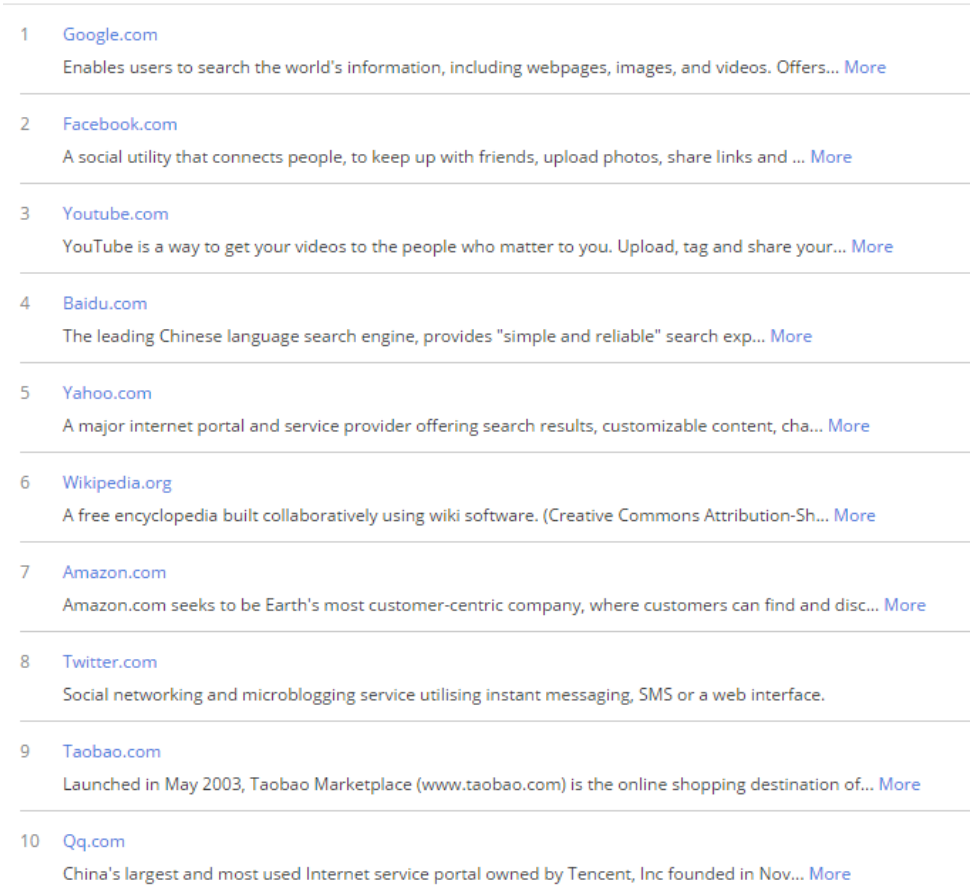

Ilustración 3 – Top Sitios Mundial según Alexa.com

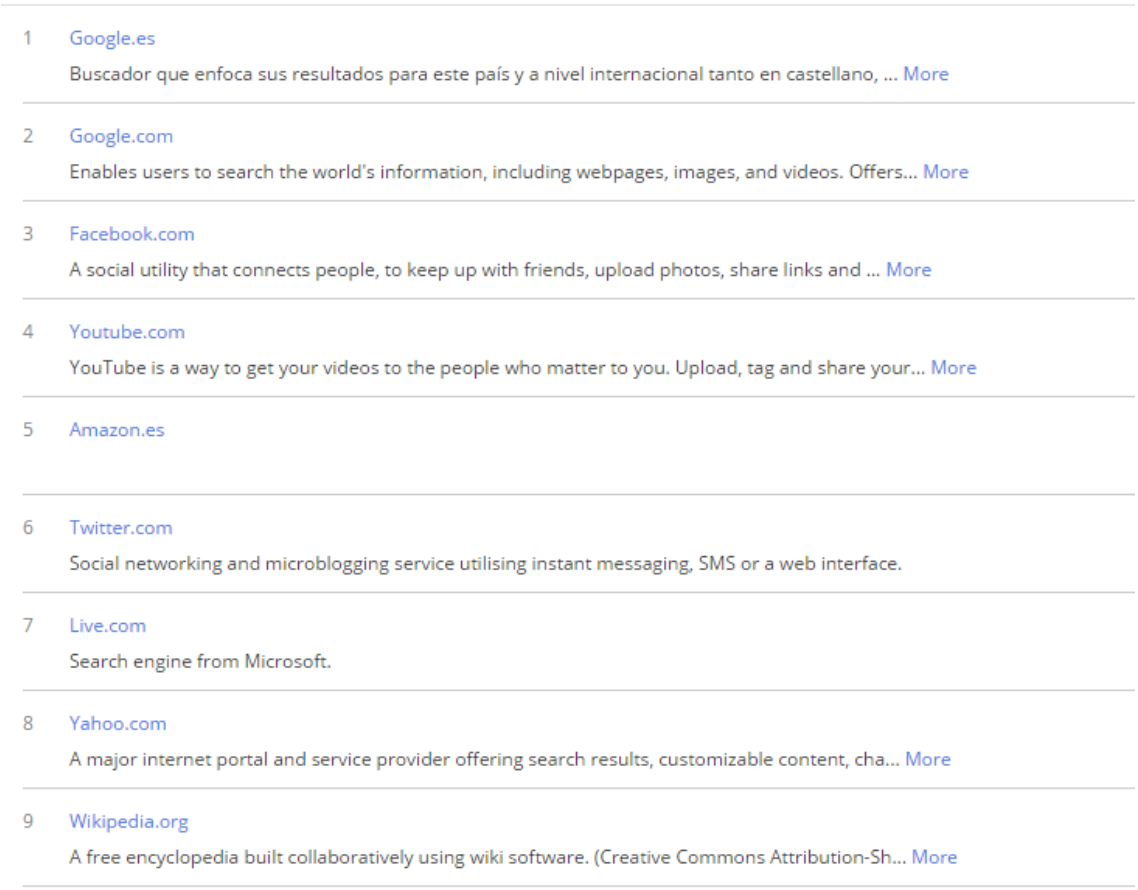

Ilustración 4 – Top Sitios en España según Alexa.com

Independientemente de sus diferentes enfoques, desde enfoques más orientados a la actividad profesional como puede ser el de LinkedIn o basados en cuestiones más personales como puede ser Facebook, todas las redes sociales tienen en común que proporcionan una herramienta muy potente de comunicación entre personas y entre personas y compañías.

Su facilidad de uso, inmediatez y actual accesibilidad desde cualquier tipo de dispositivo hacen que estén muy ampliamente extendidas y las hacen propicias para experiencias como la analizada en el presente Proyecto Fin de Carrera que finalmente se basa en el estudio del intercambio de información entre los alumnos y docentes de la asignatura a través de la red social Twitter.

2.1.2 Twitter

La Red Social Twitter apareció en el año 2006, se fundamenta en el envío y recepción de mensajes entre usuarios registrados plataforma. Tiene la característica diferencial de que los mensajes publicados no pueden superar los 140 caracteres[1].

1 Durante la realización del presente trabajo Twitter ha ampliado a 10.000 caracteres el límite para los mensajes directos (mensajes privados intercambiados entre usuarios).

A día de hoy, en apenas nueve años de andadura, Twitter presume de tener más de 280 millones de usuarios activos mensuales que intercambian 500 millones te *tweets* al día. Además está plenamente adaptada al uso de tecnologías móviles como demuestra que el 80% de los usuarios activos utilizan el móvil [11].

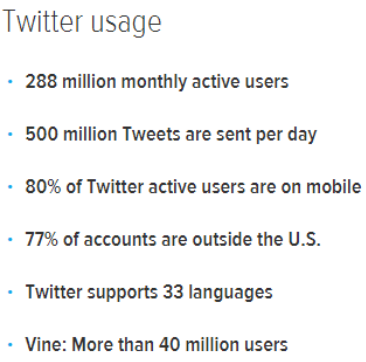

Ilustración 5 – Uso de Twitter

La limitación a 140 caracteres viene provocada por la compatibilidad inicial con los servicios de envío de mensajes de texto SMS, muy populares en la época del lanzamiento de Twitter. En la actualidad es una característica diferenciadora del servicio permitiendo a los usuarios revisar de forma rápida los asuntos de su *timeline*.

Con la ampliación actual a 10.000 caracteres en mensajes privados, directos entre usuarios, se pretende mantener por un lado la inmediatez y agilidad de la publicación con una mayor potencia en el intercambio de mensajes privados, donde se hace complicado realizar conversaciones con el límite de 140 caracteres si no es con el envío de varios mensajes para una única cuestión.

Por otra parte esta limitación ha hecho prosperar servicios como bit.ly, goo.gl o tr.im que dada una url generan otra url reducida en número de caracteres. Estas urls reducidas pueden ser incorporadas a cualquier mensaje reduciendo de esta forma su tamaño y dejando más espacio para el contenido del mensaje a la vez que permiten complementar el mensaje con un acceso externo.

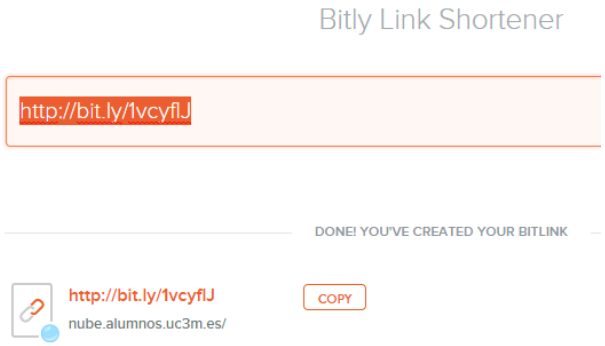

Ilustración 6 – URL acortada del servicio nube.alumnos.uc3m

A los mensajes intercambiados en Twitter se les denomina *tweets*.

El primer paso para utilizar Twitter es crearse un usuario. La creación del perfil genera una url propia del usuario dentro de la plataforma, es decir, se genera la *bitácora* de forma automática, salvando de esta forma uno de los rechazos enumerados en las Redes Sociales Indirectas. No es necesario que el usuario tenga conocimientos informáticos.

A modo de ejemplo se muestra la *bitácora* del usuario el que se basa el estudio del presente proyecto, ubicada en https://twitter.com/miisi_uc3m.

Ilustración 7 – Usuario Twitter de la asignatura

Comentábamos que la Red Social Twitter se basa en el envío y recepción de mensajes. Para facilitar y potenciar el uso de la herramienta, el propio portal sugiere otros perfiles a los que seguir, *following*. Aquellos usuarios que tienen visibilidad sobre los *tweets* que el usuario publica son sus seguidores, *followers*.

Tras la creación se pregunta al usuario sobre los temas que son de su interés. Esta característica es común a muchas Redes Sociales y pretenden romper el rechazo inicial del uso de la herramienta ya que de forma muy sencilla, al finalizar la configuración del perfil ya tiene mensajes que consultar. Lógicamente posteriormente a la creación del perfil se pueden seguir incorporando usuarios a seguir bien en base a las sugerencias que la propia herramienta sigue realizando bien indicados explícitamente.

Ilustración 8 – Sugerencias de usuarios a seguir

Todos los mensajes que emitan los usuarios que se siguen, aparecerán listados en orden inverso al momento del envío en el *Home* o *Timeline* del usuario.

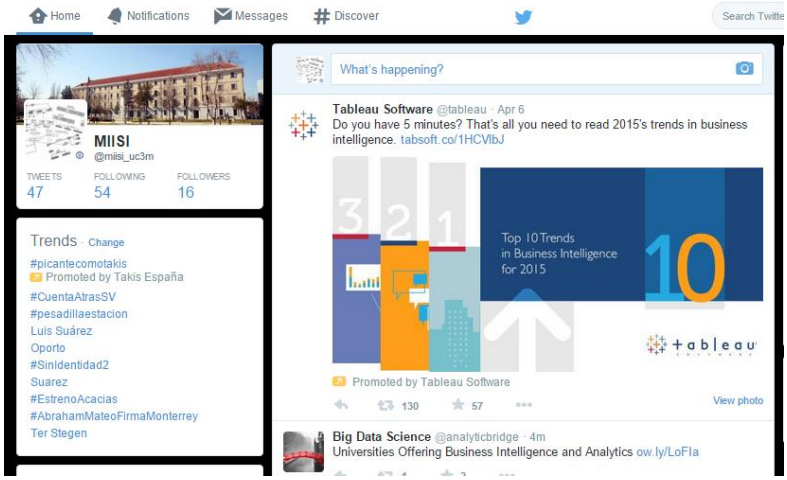

Ilustración 9 – Home del usuario

Publicar un mensaje es tan sencillo como escribirlo y enviarlo. Se incluye la posibilidad de añadir una imagen y localización. En el caso de que se quiera enviar una url, se recomienda la utilización de alguna herramienta de compactación de urls. Todas las personas que sigan al usuario (*followers)*, verán automáticamente el mensaje en su *home.*

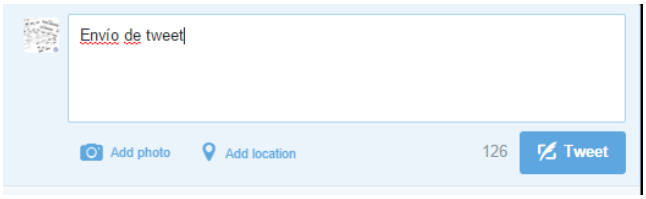

Ilustración 10 – Envío de mensajes

A este funcionamiento básico se le añaden algunas operaciones que facilitan la compartición de información, la comunicación y relación entre otros usuarios de la plataforma. De forma que:

- Un usuario puede hacer un *retweet* de un *tweet*.

  Dado un *tweet* del *home* de un usuario, este puede decidir compartirlo con sus *followers*. En el momento en el que se hace el *retweet*, el mensaje es publicado en los *home* de sus *followers*. El mensaje es marcado para diferenciarlo de los mensajes publicados originalmente por el usuario.

Ilustración 11 – retweet

- Si al hacer el *retweet* se incluye alguna modificación, estaríamos hablando de un *tweet* modificado.

  Típicamente se incluye *MT* al inicio del mismo indicando que ha sido modificado.
  Como buena práctica hay usuarios que no gustan de utilizar los 140 caracteres del mensaje dejando algunos libres para que otros usuarios puedan *retweetear* el mensaje incluyendo algún comentario.

- Menciones @.

  Para referir a un usuario se utiliza @ seguido por el nombre del usuario.
  No es necesario seguir o ser seguido por la persona referenciada.
  En la aplicación oficial, dentro de las notificaciones, existe un apartado específico para las menciones que nos han realizado.

Ilustración 12 – Menciones

- Etiquetas, Hashtag #.

  Con añadir # delante de cualquier palabra se está creando un *hashtag*. Generalmente se utiliza para marcar el contexto de la conversación en un determinado tema o cuestión.
  Esta nomenclatura resulta interesante para hacer seguimiento de todos los *tweets* que se han escrito sobre una cuestión determinada.

  Como curiosidad existen algunos *hashtags* creados por la propia comunidad de usuarios que se han extendido como *estándar de facto*.

  - #FF o #Following Friday.
    Al utilizar este *hashtag* en un *tweet* un usuario pretende recomendar a sus *followers* que sigan a otro determinado usuario.

- Treding Topic.

  No son más, ni menos, que los diez temas del momento.

Basado en la utilización del *hashtag*, Twitter aplica un algoritmo para obtener aquellos *hashtags* más utilizados en el momento. El concepto ha ido evolucionando hasta contextualizarlo en base a localización e intereses del usuario.

Dada la repercusión de Twitter, los *treding topic* pueden llegar a tener una repercusión y un impacto muy elevados, siendo frecuentemente fuente de información para los medios tradicionales.

Es muy habitual ver como cualquier tipo de publicación o evento incluye un *hashtag* en sus publicaciones para propiciar que la propia red se encargue de difundir su mensaje. Independientemente de que la cuestión llegue o no a ser *treding topic*, como se indicaba con anterioridad el *hashtag* facilita y potencia la obtención de todos los *tweets* que se hayan publicado sobre un determinado asunto.

- Mensajes Directos.

  Se puede establecer una conversación privada con cualquier usuario mediante los mensajes directos.

  Al contrario que con las notificaciones, para enviar un mensaje directo a un usuario, es necesario que el usuario sea un *follower* nuestro. Para establecer la conversación de forma bidireccional, será necesario que nosotros a su vez seamos *followers* del otro perfil para que nos pueda responder.

- Borrado.

  Se puede borrar un tweet previamente publicado. Se borrarán en cascada los retweets pero no los tweets modificados.

- Favoritos.

  Con el objeto de tener accesibles los *tweets* que consideremos de mayor interés, se pueden marcar con el atributo de favorito.
  La propia herramienta proporciona un acceso directo al listado de favoritos.

Ilustración 13 – Favoritos

- Listas.

  Otro mecanismo de organización de *tweets* son las listas. En el *timeline* se pueden seguir temas de multitud de ámbitos y usuarios de diferentes contextos (familiares, profesionales, de ocio, etc.). La creación de listas viene a dar respuesta a una necesidad de organización de las publicaciones en base a un contexto más general.

  Una curiosidad de las listas es que de la misma forma en la que se puede conocer aquellas personas que te siguen, se pueden también conocer las listas que te siguen.

2.2 Knowledge Discovery Database

"Desde el amanecer de la civilización hasta el 2003, se crearon más de 5 Exabytes de información. En la actualidad, esta cantidad se está generando cada 2 días y el ritmo sigue creciendo".

Esta afirmación era hecha en 2010 por uno de los creadores de Google, Eric Schmidt [12]. Independientemente de que la aproximación pueda ser más o menos cuestionada, es indudable que la cantidad de información que circula por Internet es de unos volúmenes imposibles de racionalizar.

Aterrizando esta afirmación en nuestro entorno más cercano e inmediato, cualquier empresa o actividad humana genera hoy en día una cantidad de información muy elevada, pero ¿de qué sirve tanta información si no somos capaces de convertirla en conocimiento?

El Knowledge Discovery Databases (KDD) persigue precisamente ese objetivo, convertir los datos en conocimiento.

Podemos distinguir entre las siguientes etapas.

- Recolección de Datos.

  Toma de Datos de las variables que se quieren analizar de las múltiples fuentes disponibles.

- Preparación y Modelización.

  Podríamos definir esta etapa como la conversión de los datos en Información. Se limpian los datos no relevantes y los relevantes se modelan en un determinado lenguaje de entendible y lógicamente adaptado al posterior proceso de minería de datos.

- Minería de datos.

  La minería de datos es una etapa consistente en dada una definición, aplicar la computación (análisis automático o semiautomático) de grandes volúmenes de información modelizados con el objetivo de obtener patrones y modelos comprensibles de representación de información no evidente para un uso posterior.

- Interpretación y Evaluación.

  Proceso de explicar y dar sentido a los resultados obtenidos.
  Dependiendo de los resultados es posible que haya que retornar a los pasos anteriores para proceder con un nuevo reprocesamiento.

- Conocimiento

  La consolidación de los resultados obtenidos como conocimiento es el objetivo final del proceso.

Con todo este proceso lo que finalmente se persigue es la obtención de Conocimiento a través de los Datos.

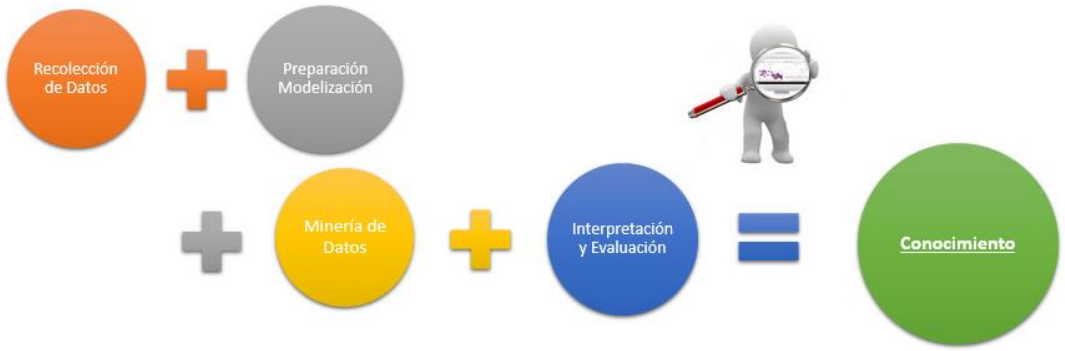

Ilustración 14 – Knowledge Discovered Databases (i)

Los datos y la información toman pues valor en la medida en la que ayuden a la compresión de un determinado problema, a la toma de decisiones o puedan ser incorporados en el proceso de análisis de otras cuestiones.

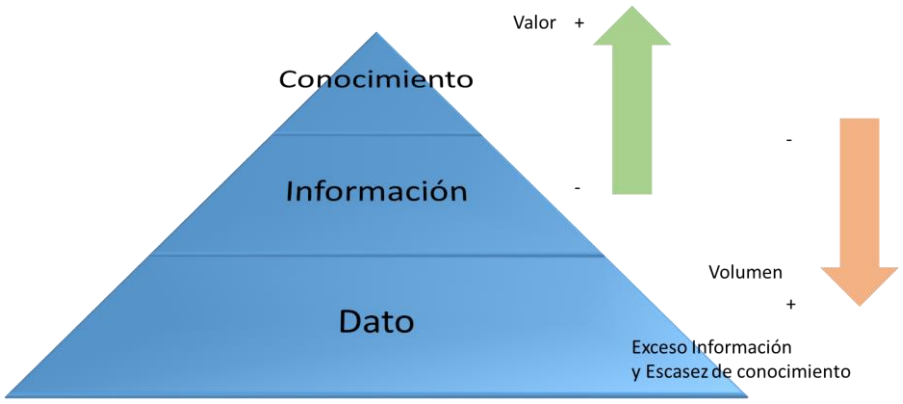

Ilustración 15 – Knowledge Discovered Databases (ii)

2.2.1 Minería de Datos

La minería de datos se fundamenta técnicamente en la Inteligencia Artificial y la Estadística, distinguiéndose las siguientes técnicas como las más comunes.

- Redes Neuronales Artificiales (RNA).

  Técnica basada en la Inteligencia Artificial. Se basa en simular artificialmente un cerebro animal, esto es, componer una red de conexión entre unidades con el objeto de obtener un resultado, estímulo final. Tiene las siguientes características:

  - Auto-organización y Adaptabilidad. Utilizan algoritmos de aprendizaje adaptativo, esto es, son capaces de aprender y adaptarse en base a los resultados obtenidos.
  - Procesado no lineal. Por su propia definición de red, cada unidad puede estar conectada a otras unidades de forma que existen múltiples caminos que se pueden llegar a seguir para conectar dos unidades. Esta capacidad permite la detección de patrones.
  - Procesado en paralelo. Al no existir una única vía de conectividad se pueden seguir varios procesamientos en paralelo.

  El nodo o unidad recibe inputs de otras unidades o del exterior. A cada input se le asocia un peso *w*. El peso puede ser alterado en el proceso de aprendizaje. Cada unidad aplica la función de activación *f*, para finalmente aplicar una función de transferencia al resultado de la función de activación con el objeto de acotar la salida *Y* de la función de activación en base a una determinada interpretación.

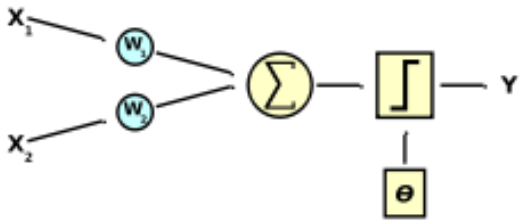

Ilustración 16 – Unidad de RNA con dos entradas

- Árboles de Decisión.

  Nuevamente se trata de una técnica proveniente de la Inteligencia Artificial. Se trata de un modelo de predicción en el que dado un conjunto de datos se construyen diagramas lógicos relacionando cada uno de los nodos en forma de árbol.

  Un árbol de decisión se recorre hacia los nodos hoja que representan la decisión a tomar. Partiendo de una determinada entrada y por medio de un conjunto de atributos se va respondiendo en cada nodo alcanzando el siguiente hasta llegar a un nodo hoja.

  Existen diferentes tipos de nodos

- Nodo decisión. Contiene un test o función a aplicar sobre algún valor de las propiedades o atributos. Típicamente representado por un cuadrado.

- Un nodo de probabilidad, que a diferencia del resto se representa de forma redonda, indica que debe ocurrir un evento aleatorio en relación con la naturaleza de la cuestión a evaluar.

- Nodo Hoja. Como se ha indicado con anterioridad, representan el final del camino y en consecuencia la decisión a tomar de seguir dicho camino.

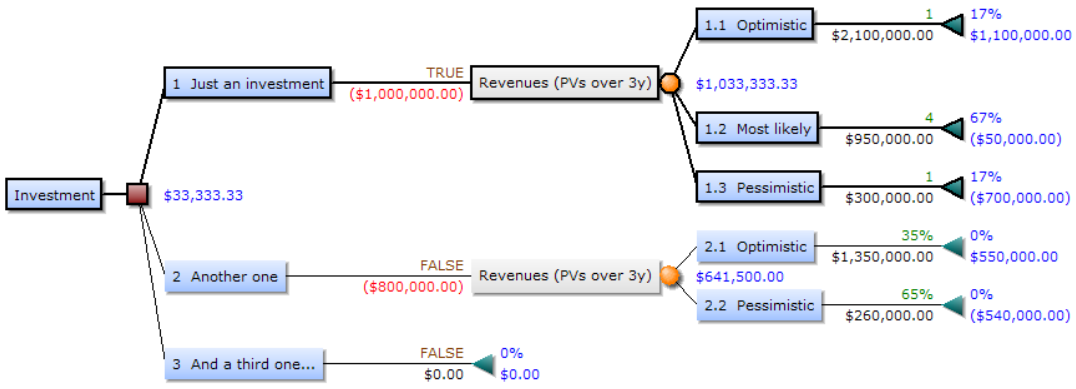

Ilustración 17 – Ejemplo Árbol Decisión

- Algoritmos Genéticos.

  Se trata de algoritmos que imitan la evolución de la especie humana con mecanismos de búsqueda de la solución basados en las leyes evolutivas de selección natural. Se inspiran en el principio de supervivencia del más apto.

  Las características principales de este tipo de algoritmos son:

  - Son algoritmos estocásticos de búsqueda múltiple.
    Es decir, dos ejecuciones diferentes pueden dar resultados diferentes.

  - La búsqueda se realiza mediante la generación objetos, de individuos de una determinada población. Estos individuos representan la solución al problema.

  - Esos individuos se descomponen o codifican en cromosomas.

  - A los cromosomas se les aplican operaciones genéticas.

    - Selección.
      Tras aplicar una función de aptitud, se determina cómo de bueno es el cromosoma, en definitiva, cómo es de buena la solución que representa ese individuo.

Los individuos que sean mejores serán los *padres* de la siguiente generación.

- Cruzamiento.
  Se produce el cruce de soluciones para dar una nueva solución. Cuando más aptas sean las soluciones seleccionadas, mejor será en teoría el nuevo individuo.

- Mutación.
  Con el objeto de ampliar el campo de búsqueda de la solución e intentar cubrir soluciones no planteadas a priori, se producen cambios aleatorios sobre los cromosomas de forma que se generan nuevos individuos que pueden haber sido obviados del conjunto de individuos original.

- Remplazo.
  Una vez determinado el operador genético, se seleccionan los mejores individuos y se descartan los que no cumplen el umbral elegido. Se produce una transición componiendo una nueva población de individuos.

- *Clustering* o Agrupamiento.

  No se pretende determinar el valor de una variable sino que estas técnicas consisten en crear agrupaciones entre instancias con características similares. El objetivo es la propia estructura y agrupación de datos.

  En estas técnicas se parte de objetos no etiquetados, son elementos que no parten de una categorización o evaluación previa como ocurría en las técnicas anteriores. Estamos hablando de aprendizaje no supervisado.

  Los objetos tienen un conjunto de características, las técnicas de c*lustering* o agrupamiento consisten en evaluar las características de cada elemento e ir generando agrupaciones de elementos con características similares. Hay que tener en cuenta que dada la heterogeneidad que pueden presentar los datos, las propiedades de los mismos pueden ser dispares o estar expresadas en conceptos diferentes con lo que la etapa de transformación cobrará especial relevancia.

  Su aplicación en minería de datos es clara ya que permite reconocer patrones y generar modelos.

  Existen diferentes modelos de agrupamiento, cada uno utiliza un tipo de algoritmos para generar sus *clústeres*, algunos de ellos son:

  - NN (Nearest Neighbor).
    Dado un conjunto de individuos en un espacio, se conecta con los individuos más cercanos, cada grupo de elementos conectados, define un *clúster*.

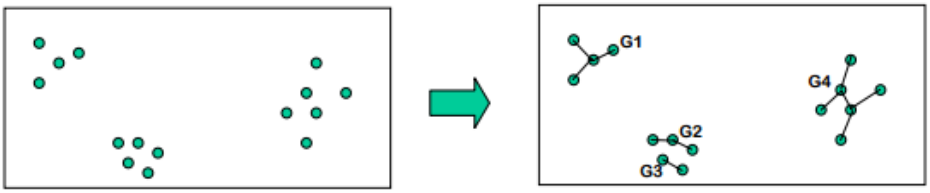

Ilustración 18 – NN (Nearest Neighbor) Clustering

- k-medias.
  Se tienen que formar k grupos minimizando la distancia entre los elementos de cada grupo. Se eligen k elementos como semillas. Se van enlazando el resto de elementos a la semilla más cercana.

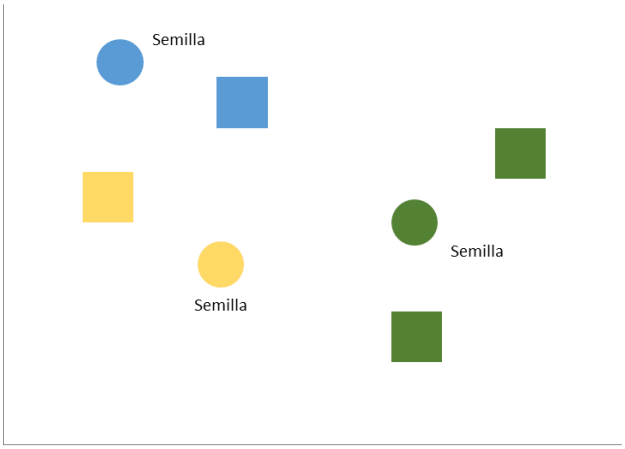

Ilustración 19 – k-medias - Inicio

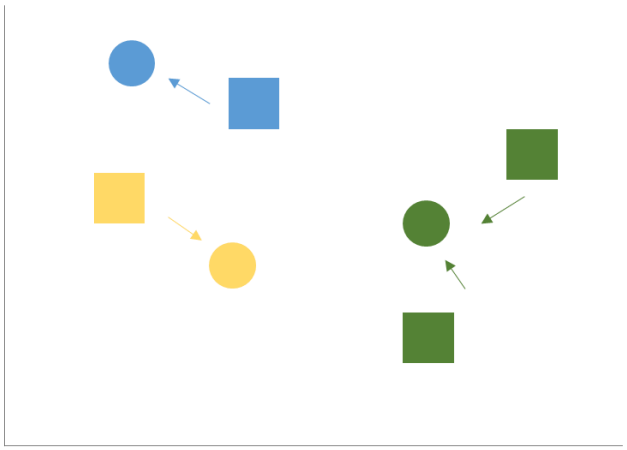

Ilustración 20 – k-medias – Inicio conectado

Se calcula la media y el punto que sale representa la nueva semilla.

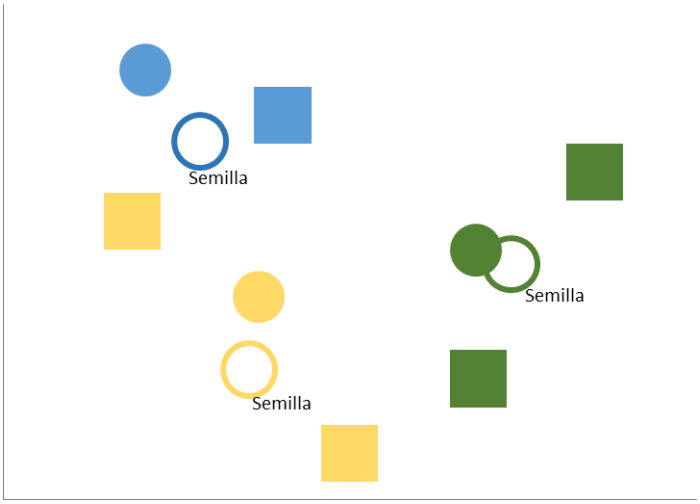

Ilustración 21 – k-medias – Interacción 1

Sobre las nuevas semillas se recalculan las conexiones. Es posible que con la nueva organización algún elemento cambie de grupo.

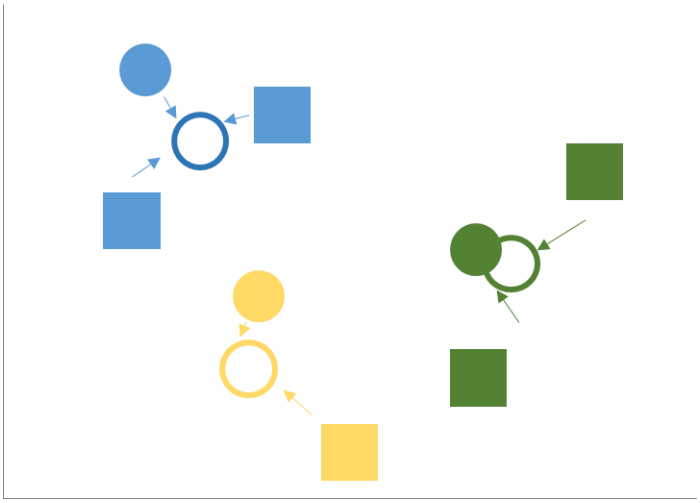

Ilustración 22 – k-medias – Interacción 1 conectada

La operación se va repitiendo hasta que los centros no varíen.
Dado que los centros van variando, como ha podido comprobarse, los elementos pueden cambiar de grupo. Al finalizar todas las interacciones, se obtiene la agrupación definitiva.

Un problema de este método es que el resultado depende en gran medida de las semillas elegidas al inicio.

2.2.2 Aplicaciones de KDD

Para finalizar con este apartado, se enumeran algunos ejemplos de aplicaciones prácticas de la minería de datos.

- Aplicación en áreas de Marketing y Venta

  Mediante la recolección y análisis de los datos obtenidos de los hábitos de compra, la consulta de sitios web o de algún producto o conjunto de productos, el uso de tarjetas de fidelización, las compañías diseñan acciones de marketing y recomendaciones de compra adaptadas al individuo o al conjunto en el que se englobe un determinado individuo.

- Aplicación en cuestiones relativas a la seguridad

  Desde la detección de patrones de fraude dentro de una compañía, por ejemplo recopilando la actividad de consulta en ciertas aplicaciones, hasta la trazabilidad de un individuo en la red, pueden determinarse patrones de comportamiento y detectar cuestiones dignas de analizar desde el punto de vista de la seguridad.

- Aplicación científica

  Partiendo de la enorme generación de datos obtenida por ejemplo de las sondas espaciales se puede llegar al descubrimiento de objetos estelares nuevos.
  Otro ámbito científico de actuación clásico son los modelos de predicción meteorológica.

2.3 Minería de Sentimientos

“Si me preguntaran sobre la revolución que se nos viene encima y que nos va a desconcertar a todos, respondería, sin vacilar, la irrupción del aprendizaje social y emocional en nuestras vidas cotidianas.” [13]

Esta afirmación de Eduard Punset recoge el interés que hay en los últimos tiempos por el desarrollo de lo que se ha denominado inteligencia emocional.

Entendemos por inteligencia emocional el desarrollo de habilidades de expresión y modulación de las emociones propias así como el entendimiento de las emociones de las personas que nos rodean y la aplicación de ese conocimiento a nuestro comportamiento. [14]

Las aplicaciones de la minería de sentimientos son múltiples y están muy de actualidad, por ejemplo en el campo del marketing y la publicidad para medir el posicionamiento de una marca en el mercado, cómo perciben los clientes la marca o el efecto de una determinada campaña comercial.

Es precisamente este tipo de minería en el que se fundamenta el proyecto.

La minería de sentimientos se basa en combinar las técnicas de Procesamiento del Lenguaje Natural (PLN o Natural Language Processing NLP en inglés) con técnicas de análisis de opiniones y sentimientos. Digamos

que con el primer proceso se logra la organización y clasificación de la información en base a criterios lingüísticos permitiendo en una segunda fase un análisis desde el punto de vista de las opiniones y sentimientos.

En el desarrollo del proyecto se han combinado estas técnicas, de forma que tras las tareas de extracción y procesamiento de la información de la cuenta de Twitter de la asignatura, se han aplicado técnicas de procesamiento de lenguaje natural para la definición de una ontología sobre la que, con la ayuda de los patrones y relaciones entre expresiones, se pretende llegar detectar los sentimientos y opiniones expresadas por los alumnos.

Capítulo 3

Análisis y Diseño

Con el marco de trabajo establecido en el estado de la cuestión, teniendo en cuenta las tecnologías disponibles y sin perder de vista los objetivos del proyecto, en este apartado se realiza un análisis de las labores a ejecutar y el diseño de la implementación.

3. Análisis y Diseño

En la ejecución de todo proyecto es fundamental abordar una fase de análisis y diseño entrando en profundidad en cada uno de los elementos del proyecto.

El apartado aborda los siguientes puntos.

- Metodología.

  Se explica la metodología seguida en la ejecución del proyecto.

- Recolección de Datos.

  Se analizan las tecnologías en las que se basa el API de Twitter así como las herramientas que se van a utilizar en la fase de implementación.

- Transformación/Preparación.

  Partiendo de los datos extraídos de Twitter se diseña una aplicación para la conversión de los datos en lenguaje natural, con el objetivo de su posterior procesamiento.

- Minería de sentimientos.

  A partir de la información obtenida en la fase de Transformación/Preparación en este apartado se describe el uso de la herramienta knowledgeMANAGER para la creación de la ontología y ejecución del análisis de la información.

- Entorno tecnológico.

  Se dispone esquemáticamente el entorno tecnológico necesario para le ejecución de cada una de las tareas anteriores.

3.1 Metodología

La metodología seguida en la implementación del presente Proyecto Fin de Carrera sigue los pasos habituales de los trabajos de KDD.

- ¿Qué queremos medir? ¿En qué toma de decisión o mejora de proceso se quiere aportar conocimiento?

  Para iniciar los trabajos, conviene detenerse y repasar cuál es el objetivo del trabajo, qué es lo que se quiere medir.

  Minería de sentimientos sobre Twitter.

El Proyecto se enmarca dentro de las acciones de innovación docente provocada por la necesidad de mejora continua basada en el establecimiento de canales de comunicación entre docentes y alumnos.

Mediante el uso de un servicio de *microblogging* como es Twitter, se facilita la participación y acceso de todos los integrantes del grupo de la asignatura desde el inicio del curso.

Con el análisis de los mensajes y sentimientos expresados en la cuenta de Twitter creada al efecto se pretende disponer de una retroalimentación de la labor docente de cara a implementar un proceso de mejora continua.

- Muestreo. Recolección de Datos.

El presente Proyecto Fin de Carrera basa su estudio en los mensajes y sentimientos de la cuenta Twitter de la asignatura creada para este propósito de mejora continua mediante la Innovación Docente: miisi_uc3m - *https://twitter.com/miisi_uc3m*

Ilustración 23 – Cuenta Twitter de la asignatura

Los docentes utilizan el *blog* para publicar información o generar temas de debate relacionados con el desarrollo de la asignatura.

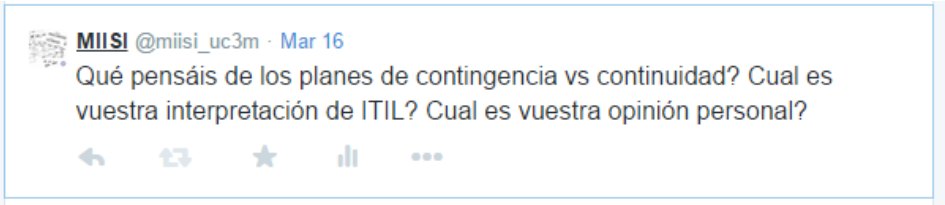

Ilustración 24 – Ejemplos de Tweets del equipo docente

Por su parte los estudiantes utilizan el medio para plantear cuestiones o comentar aspectos de la asignatura.

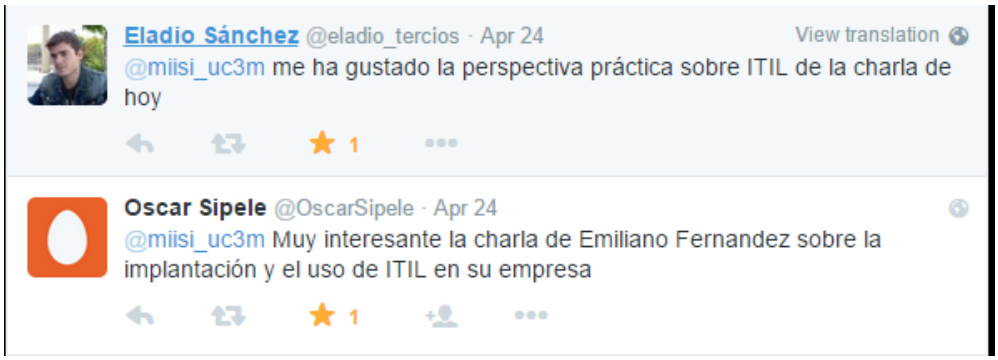

Ilustración 25 – Ejemplos de Tweets de los alumnos

Mediante el uso del API de Twitter y apoyados en una herramienta pública accesible desde la propia página de Twitter se extraen todos los *tweets.*

- Transformación y preparación de la información

El API de Twitter devuelve los datos en forma de objetos con multitud de propiedades (creador, fecha y hora, identificador, propiedades de localización del envío, etc.). Adicionalmente el formato en el que se organiza el objeto no es directamente entendible o traspasable a ninguna herramienta de PLN.

Es por eso que se ha hecho necesaria el diseño y desarrollo de una herramienta de procesamiento y tratamiento de la información para traducirla a un formato que permita trabajar con ella en las fases posteriores.

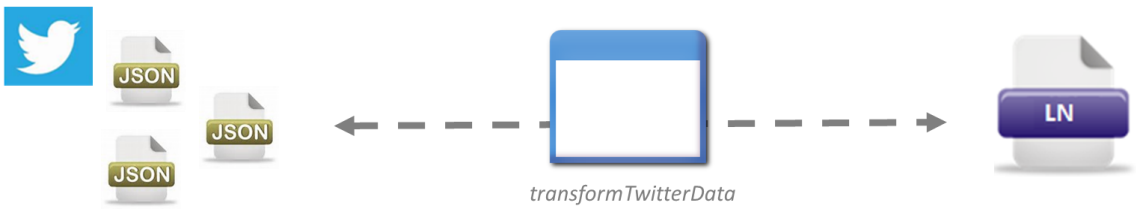

Ilustración 26 – Objeto del programa de transformación

- Minería de sentimientos.

  Apoyándose en la herramienta knowledgeMANAGER se ha desarrollado una ontología que guíe el desarrollo del proceso de minería de sentimientos.

  A partir de la información obtenida de los pasos anteriores se ha desarrollado una conceptualización de los *tweets* de la asignatura clasificándolos en base a su tipología, elaborando patrones y estableciendo relaciones que permiten su estudio desde el punto de vista de los sentimientos.

- Interpretación y Evaluación.

  Todos los pasos anteriores van encaminados a la obtención de conocimiento.
  El modelo generado no es más que la herramienta que permitirá la indexación de los *tweets* actuales (y potencialmente de los *tweets* futuros) para la obtención de conocimiento.

3.2 Recolección de Datos

Como ya se ha comentado con anterioridad el estudio se basa en la información intercambiada en la cuenta de Twitter de la asignatura (@miisi_uc3m).

Es por tanto que el proceso de selección de datos del proceso de minería no se basa en técnicas aleatorias o con un pre-procesamiento sobre un gran volumen de datos sino que se basa en la obtención de toda la información intercambiada en la famosa Red Social.

3.2.1 Requisitos Funcionales

Existe por lo tanto dos requisitos funcionales claramente definidos.

Requisito	**RF-001**
Nombre	Recolectar *tweets* en un momento del tiempo.
Descripción	El proyecto se basa en la minería de los *tweets* intercambiados en el perfil de la asignatura. Por la naturaleza del proyecto, no es necesario mantener una actualización constante sino que a ser posible es más sencillo poder recuperar todos los *tweets* en un determinado momento del tiempo, al final del curso.
Necesidad	Alta
Dependencias	Dependencia de los APIs que proporcione Twitter.

Tabla 2 – Recolección de datos RF-001 – Recolectar tweets en el tiempo

Requisito	**RF-002**
Nombre	Recolectar todos los *tweets.*
Descripción	El proyecto no pretende estudiar los *tweets* de una determinada tipología, de un usuario en concreto, de una temática, etc. sino que el objetivo es poder tratar todos los *tweets*.
Necesidad	Alta
Dependencias	Dependencia de los APIs que proporcione Twitter.

Tabla 3 – Recolección de datos RF-002 – Recolectar todos los tweets

En ambos requisitos existe una dependencia de los APIs que proporciona Twitter, es sin duda una dependencia fundamental. Pasemos a analizar las posibles alternativas.

3.2.2 APIs de Twitter

Twitter ofrece tres APIs claramente diferenciados: Streaming API, REST API y Search API. Dos de ellos se implementan con tecnología REST (REST API y Search API) mientras que el Streaming API está orientado a proporcionar *tweets* en tiempo real [15].

Veamos a continuación las características de cada API de cara a la elección del más oportuno para proyecto.

- Streaming API [16] [17]

  Es un API orientado mostrar información en tiempo real. Proporciona a los desarrolladores acceso con baja latencia al flujo global de datos de Twitter. Las aplicaciones que lo implementen harán *pooling* [18] para obtener los el estado de los objetos que se estén filtrando.

  Proporciona el acceso a los estados de todos los usuarios públicos, filtrados por diferentes criterios (por ID de usuario, palabra clave, ubicación geográfica, etc.)

  A modo de ejemplo este API es utilizado por aplicaciones como TweetDeck.

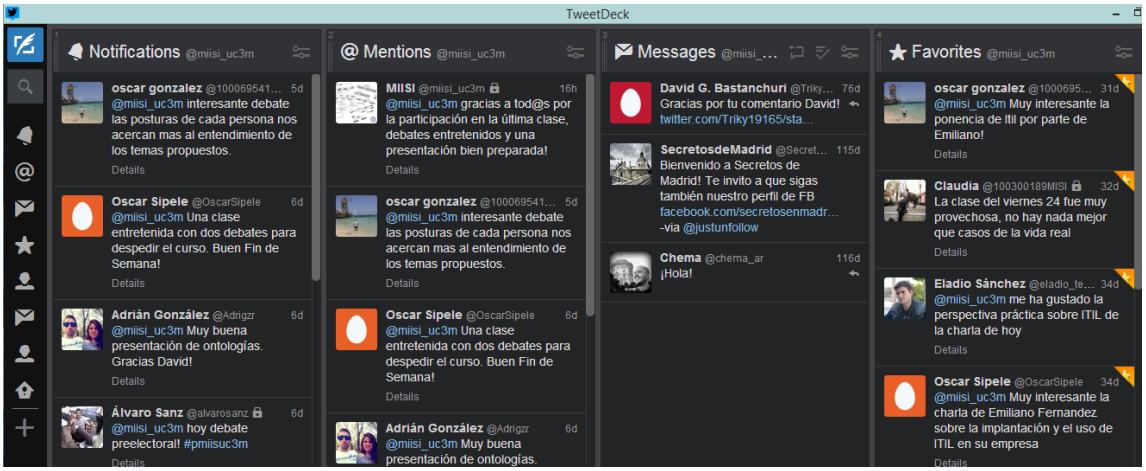

Ilustración 27 – Aplicación usando Streaming API - Tweet Deck [19]

Este API retorna los resultados en formato *json* [20].

- Search API [17] [21]

  Su utilidad es la de proporcionar un método para realizar búsquedas de *tweets* a partir de una determinada palabra clave o patrón determinado.

  Tiene las siguientes características.

- No se pueden buscar tweets de cualquier momento, sólo se muestran los más recientes. El API no pretende ser una herramienta de búsquedas masivas de *tweets* sino que está focalizada en la relevancia

- No se muestran *tweets* más antiguos a una semana

- Forma parte del REST API

- Como todos los APIs de Twitter, a partir de la versión 1.1, exige estar autenticado.

La búsqueda se realiza mediante la construcción de la url de petición siguiendo un determinado formato.
La petición HTTPS se compone:

https://api.twitter.com/1.1/search/tweets.json?q=<palabra_clave>

En el siguiente ejemplo podemos ver cómo se ejecutaría una consulta para obtener los últimos *tweets* publicados en el perfil uc3m.

https://api.twitter.com/1.1/search/tweets.json?q=uc3m

Petición: Como puede verse, la sesión está autenticada utilizando tecnología OAuth. Detallada en el punto 3.2.2: OAuth

```
GET /1.1/search/tweets.json?q=uc3m HTTP/1.1
Authorization:
        OAuth
        oauth_consumer_key="DC0sePOBbQ8bYdC8r4Smg",oauth_signature_method="HMAC-
        SHA1",oauth_timestamp="1432850771",oauth_nonce="3218417271",oauth_version="1.0",
        oauth_token="3010797699-
        3LvfP1Z2EICOZ9JUQioFMpzyVZiHNvLOwct5o16",oauth_signature="pNjhZD%2BY3%2
        By2OU2%2Bpb5x%2BvJodzE%3D"
Host:
        api.twitter.com
X-Target-URI:
        https://api.twitter.com
Connection:
        Keep-Alive
```

Respuesta. En formato *json.* Se muestra un fragmento a modo ilustrativo.

```
HTTP/1.1 200 OK
x-frame-options: SAMEORIGIN
content-type: application/json;charset=utf-8
x-rate-limit-remaining:179
last-modified: Thu, 28 May 2015 22:06:11 GMT
```

```
status: 200 OK
x-response-time: 76
date: Thu, 28 May 2015 22:06:11 GMT
Connection: keep-alive
x-transaction: 0e95b597c166afd0
pragma: no-cache
cache-control: no-cache, no-store, must-revalidate, pre-check=0, post-check=0
x-connection-hash: 87112656af979e98149f1559906fcde4
x-xss-protection: 1; mode=block
x-content-type-options: nosniff
x-rate-limit-limit: 180
expires: Tue, 31 Mar 1981 05:00:00 GMT
set-cookie: lang=en
set-cookie: guest_id=v1%3A143285077177327377; Domain=.twitter.com; Path=/; Expires=Sat, 27-
May-2017 22:06:11 UTC
content-length: 66220
x-rate-limit-reset: 1432851671
content-disposition: attachment; filename=json.json
server: tsa_b
x-twitter-response-tags: BouncerCompliant
strict-transport-security: max-age=631138519
x-access-level: read-write-directmessages

{
 "statuses": [
  {
   "metadata": {
    "iso_language_code": "es",
    "result_type": "recent"
   },
   "created_at": "Thu May 28 22:05:23 +0000 2015",
   "id": 604045833315700700,
   "id_str": "604045833315700737",
   "text": "RT @HETradio3: Mañana .@toundra tocan en la .@uc3m de Leganés y nosotros ofrecemos entradas
dobles. ¿Las quieres? Envía un mail a hoyempiez…",
   "source": "<a href="http://twitter.com/download/iphone" rel="nofollow">Twitter for iPhone</a>",
   "truncated": false,
   "in_reply_to_status_id": null,
   "in_reply_to_status_id_str": null,
   "in_reply_to_user_id": null,
   "in_reply_to_user_id_str": null,
   "in_reply_to_screen_name": null,
   "user": {
    "id": 143734473,
    "id_str": "143734473",
    "name": "Nooirax Producciones",
    "screen_name": "nooirax",
    "location": "Madrid",
    "description": "Experimental rock music independent label, booking and promotions",

    "url": "http://t.co/3rf0dDQzHR",


    ….
```


- REST API [17] [22]

Provee primitivas para el acceso a leer y escribir los datos de Twitter. Crear un nuevo *tweet*, leer un nuevo *tweet* con todas sus propiedades e información adjunta (autor, fecha y hora, menciones, *hashtags*, etc.).

Al igual que los APIs anteriores, las sesiones se autentican y autorizan con OAuth y las respuestas se realizan con *json.*

Sin embargo tiene notables diferencias con los anteriores. Para empezar no necesita de disponer de una sesión abierta como ocurre con el *Streaming* API, sino que su uso puede ser puntual, en cualquier momento del tiempo.

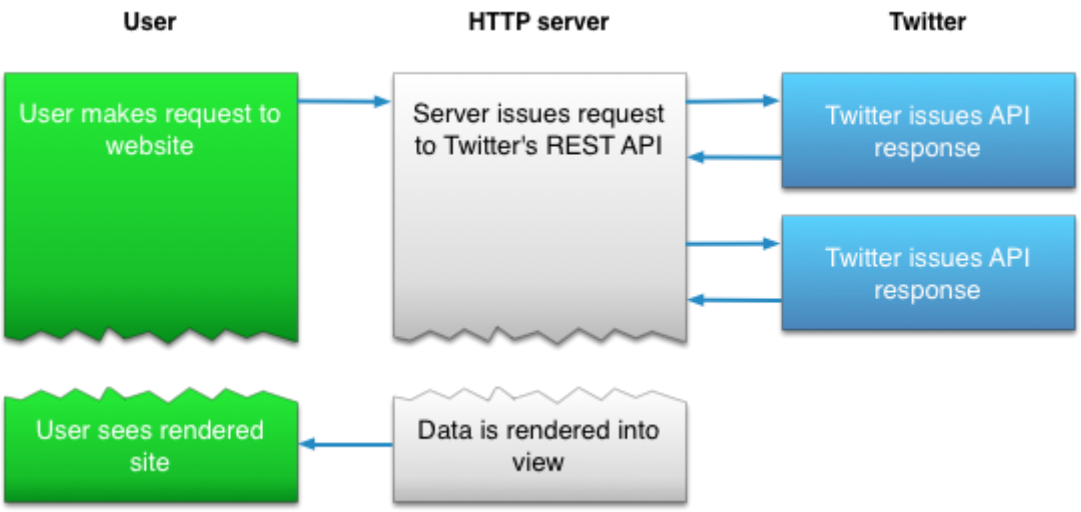

Ilustración 28 – REST API [16]

El *Streaming API* por el contrario mantiene una sesión persistente abierta contra el servidor. Mientras la sesión está abierta, realiza constantemente peticiones a Twitter para obtener el estado de los objetos filtrados.

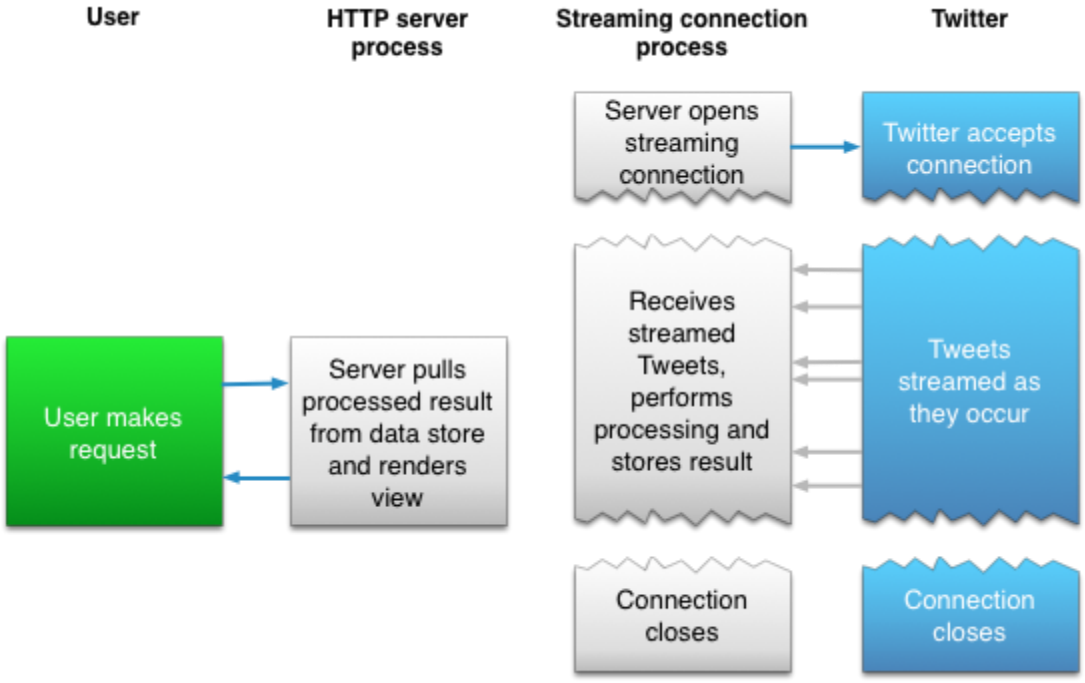

Ilustración 29 – Streaming API [16]

Esto en cuanto a las diferencias con el *Streaming* API, con respecto al *Search* API existe otra diferencia notable y es que como se ha indicado el *Search* API está orientado a búsquedas con patrones o palabras claves mientras que el REST API puede obtener de manera sencilla todos los *tweets* de una sola petición.

Parece claro por lo tanto que el REST API es el API más oportuno para los objetivos de recolección de datos establecidos.

API	**RF001 - Recolectar *tweets* en un momento del tiempo.**	**RF002 - Recolectar todos los *tweets*.**
Streaming API	Exige tener una sesión constantemente abierta e irlos almacenando localmente.	
Search API	Retorna sólo *tweets* de la última semana.	Exigiría múltiples consultas.
REST API	Puede ejecutarse en cualquier momento.	No hay limitación temporal. Sólo hay limitaciones por cantidad de peticiones por minuto que en cualquier caso son muy superiores a lo demandado para el proyecto.

Tabla 4 – Recolección de datos RF-002 – Recolectar todos los tweets

3.2.3 API REST Twitter

Continuando con el análisis y de cara a determinar el mejor método de acceso al API, se detallan a continuación el API REST de Twitter así como las tecnologías de base en las que se sustenta.

3.2.3.1 Tecnología REST

El concepto REST (Representational State Transfer) surge originalmente como un conjunto de principios de arquitecturas web (RESTful architecture) más que a una tecnología de APIs. [23] [19] [24].

Se basa en diseñar la arquitectura de un servicio de forma que se facilite el acceso a los recursos del mismo utilizando las ventajas del protocolo HTTP.

Es por eso que el término REST se utiliza para referirse a APIs basados en interfaces HTTP. Un recurso es manejado mediante órdenes HTTP simples (GET, POST, etc.)

El término surge en el año 2000 en la tesis doctoral de Roy Fielding, uno de los principales autores de la especificación de HTTP y actualmente es ampliamente utilizado como demuestra el API de un servicio de tanta popularidad como Twitter.

Los servicios web entendidos como HTTP+XML=SOAP están orientados a ofrecer un conjunto de operaciones mediante un interfaz de servicio definido en WSDL.

En contraposición a este enfoque de RPC (Remote Procedure Call), REST está orientado al acceso a recursos concretos de un servidor apoyándose en tecnologías estándar en la web y con muchos años de madurez.

- Utilización de HTTP [25].
    - Utilización de métodos estándares como POST, GET, PUT y DELETE
    - Aprovechamiento de las características de protocolo sin estado.
- Utilización de URI a la hora de localizar cualquier recurso.

Existen cinco principios clave a la hora de elaborar un servicio REST.

1. El universo del servicio se representa como un recurso. Todo es un recurso.

    Para Twitter podemos encontrar los siguientes objetos.

    - *timeline*
    - *tweet (llamados status)*
    - *trends*
    - Búsquedas almacenadas
    - Favoritos
    - Mensajes directos
    - *followers*
    - etc.

    Cada objeto dispone de una serie de recursos asociados.

    - *timeline*

        - *timeline* de usuario
        - menciones
    - *tweet*
        - *tweets*
        - *retweets*
    - *trends*
        - *trends* cercanos
    - Favoritos
        - Listado de favoritos
    - Seguidores
        - Un seguidor en concreto
        - Solicitudes entrantes
        - Solicitudes emitidas
    - etc.

2. Cada recurso tiene un identificador único.

   Una de las características de los servicios REST es que se apoyan en tecnología web estándar. En este caso los identificadores se representan con una URI [26].

Recurso	Identificador
Menciones	https://api.twitter.com/1.1/statuses/mentions_timeline.json
timeline de usuario	https://api.twitter.com/1.1/statuses/user_timeline.json
Un *tweet* concreto	https://api.twitter.com/1.1/statuses/show.json?id=210462857140252672
Un r*etweet* concreto	https://api.twitter.com/1.1/statuses/retweets/509457288717819904.json
trends cercanos	https://api.twitter.com/1.1/trends/place.json?id=1
Lista de favoritos del usuario escogido	https://api.twitter.com/1.1/favorites/list.json?count=2&screen_name=episod
Lista de seguidores	https://api.twitter.com/1.1/followers/ids.json
Peticiones de seguimiento entrantes	https://api.twitter.com/1.1/friendships/incoming.json
Peticiones de seguimiento emitidas	https://api.twitter.com/1.1/friendships/outgoing.json

Tabla 5 – Ejemplos peticiones REST Twitter

3. Se utilizan interfaces simples y estándar.

Estándar	Descripción
HTTP – GET	Método utilizado para recoger un recurso como un *tweet*
HTTP – PUT	Método utilizado para la creación y actualización de recursos
HTTP – DELETE	Borrado de recursos
HTTP – POST	Una petición de operación. Por ejemplo el envío de un *tweet*

Tabla 6 – Interfaces HTTP en REST

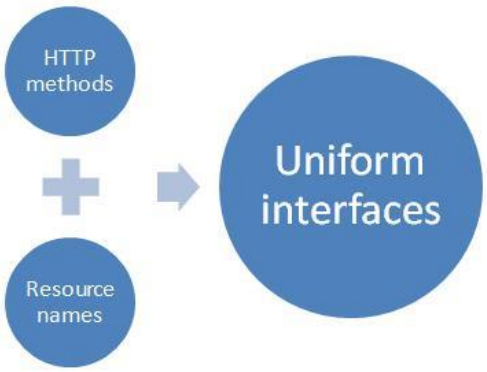

Ilustración 30 – Interfaces uniformes REST [19]

La combinando el uso de URI para representar los elementos con el uso de métodos HTTP estándar proporciona la posibilidad de hacer accesibles los servicios REST mediante un interfaz uniforme y estándar.
Podemos ver cómo se combinarían ambos con algunos ejemplos del API REST de Twitter.

Servicio	REST URI
Obtener Menciones	GET /1.1/statuses/mentions_timeline.json
Mostrar un *tweet*	GET /1.1/statuses/show.json?id=210462857140252672
Enviar un mensaje directo	POST /1.1/direct_messages/new.json
Reportar SPAM	POST /1.1/users/report_spam.json
Modificar perfil	POST 1.1/account/update_profile.json

Tabla 7 – Peticiones REST en Twitter

4. Un recurso puede tener representaciones alternativas.

   Se permite por ejemplo que clientes y servidores negocien el tipo de representación a usar en una determinada comunicación.
   Por ejemplo un mismo servicio puede retornar un elemento en HTML para que sea correctamente representado por un navegador o en XML o json para una aplicación Microsoft .NET.

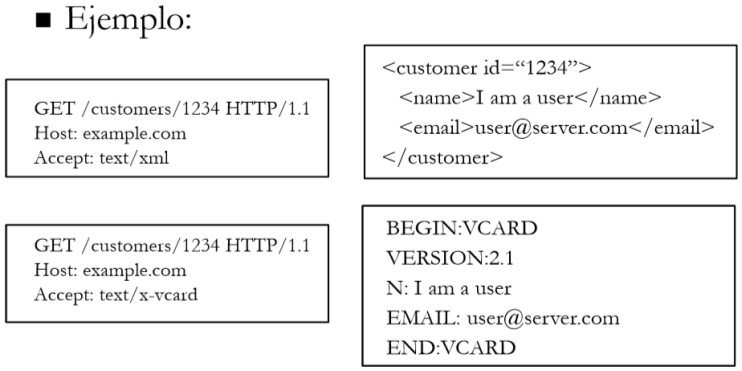

Ilustración 31 – Múltiples representaciones de un elemento REST [24]

5. No mantener estado.

   Cada petición es independiente de la anterior y no representa nada para la siguiente; no se envía ningún dato en la petición que represente el estado de la misma.

   De esta forma se consigue que en las arquitecturas basadas en servicios REST se puedan aplicar de forma sencilla técnicas de balanceo y sean muy escalables. Al no haber ningún contexto, cualquier servidor es capaz de responder una petición de cliente.

3.2.3.2 Tecnología OAuth

El API de Twitter utilizado en el proyecto es público, pero para su uso exige estar previamente autenticado y autorizado.

La tecnología que se utiliza para que una aplicación pueda utilizar el API de Twitter es OAuth (*Open Authorization)* [27].

OAuth permite el acceso de aplicaciones de terceros a servicios sin necesidad de que las aplicaciones de terceros conozcan las credenciales del usuario en el servicio.

Adicionalmente, OAuth no es sólo un protocolo para la autenticación sino que también es un protocolo mediante el que se puede gestionar autorización. Dentro del protocolo de negociación de credenciales, el usuario puede determinar que operaciones son aquellas sobre las que da acceso a las aplicaciones de terceros de forma que la aplicación de terceros no sólo no tiene los datos de conexión del usuario sino que durante el periodo de tiempo que tiene acceso al servicio, sólo tendrá acceso a aquellas opciones sobre las que el usuario le haya concedido acceso [28].

OAuth es un estándar abierto por lo que cualquiera puede implementarlo. Aunque muchos de los servicios más utilizados de Internet utilizaban sus protocolos propietarios como Facebook con Facebook Connect, en la actualidad han ido migrando a OAuth que se ha convertido en el estándar de facto al ser utilizado por muchos y muy populares servicios de Internet: Amazon, Evernote, Facebook, Flickr, Foursquare, GitHub, Google, Instagram, Intel Cloud Services, LinkedIn, Microsoft, Netflix, Paypal, Yelp, etc. [29] [30] Por su parte Twitter no sólo ha apostado por este modelo sino que también ha participado activamente en la confección de la versión 1.0.

La posibilidad de autorización convierte a OAuth una tecnología ideal para su uso por aplicaciones de terceros que explotan el API de un determinado servicio. Como puede ocurrir con cualquier cliente Twitter como por ejemplo TweetDeck o el aplicativo utilizado en el presente Proyecto Fin de Carrera para la extracción de tweets: Apigge`s API Console [31]

En una arquitectura tradicional cliente/servidor el modelo de autenticación está basado en credenciales de usuario que dan al usuario acceso a sus recursos. Típicamente el usuario en un formulario o página web introduce sus credenciales que se validan contra el servidor y se establecen los accesos oportunos.

La relación de confianza involucra dos actores:

- El usuario.

  Es la persona que consume el servicio y que se autentica para su explotación.

- El proveedor del servicio.

  Ofrece el servicio previa autenticación y almacena las credenciales y el perfil de autorizaciones de un determinado usuario.

Las siguientes gráficas ilustran el intercambio de información tradicional.

1. Un usuario accede a un determinado servicio.
   De inicio el usuario no tiene acceso, salvo que se autentique. Tradicionalmente con un usuario y contraseña que se almacenan en los recursos del propio servicio.

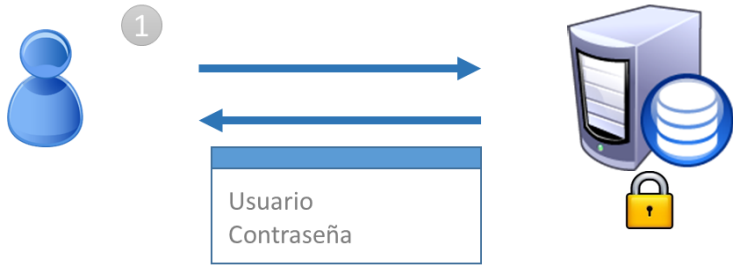

Ilustración 32 – Modelo autenticación tradicional - Acceso a la aplicación.

2. El usuario se valida contra el servicio.
   Este confirma la veracidad de los datos y el usuario queda autenticado.

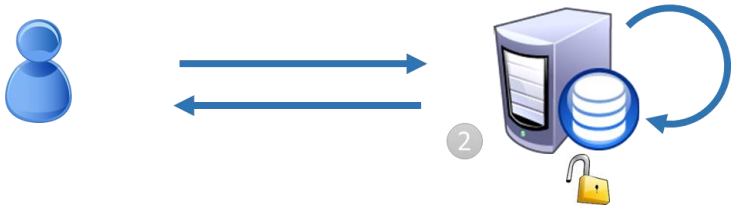

Ilustración 33 – Modelo autenticación tradicional. Validación del usuario dentro del servicio

3. Utilización del servicio.

   Una vez autenticado, el usuario accede a los servicios y operaciones sobre las que su perfil de usuario tiene permisos. Todo el esquema de validación se realiza en la propia arquitectura del proveedor de servicios.

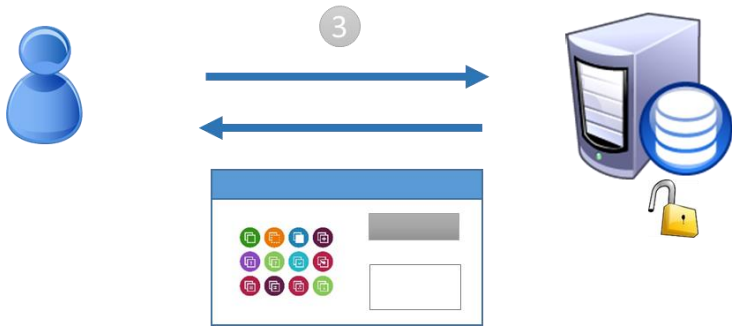

Ilustración 34 – Modelo autenticación tradicional. Utilización del servicio

OAuth surge por la necesidad de simplificar el acceso de los usuarios a los múltiples servicios de Internet.

Tradicionalmente nos encontramos con múltiples servicios cada uno manteniendo e implementando su propio sistema de autenticación y autorización. Sistemas de autenticación y autorización que por un lado se asemejan al modelo tradicional mostrado, pero que por otro lado incorporan particularidades del servicio que los convierten en incompatibles unos con otros.

Finalmente un usuario se ve obligado a mantener múltiples perfiles de usuario, posiblemente cada uno de ellos con un nombre y una contraseña diferente debido a la heterogeneidad de las políticas de mantenimiento de credenciales; dificultad para tener un nombre de usuario reconocible, diferentes políticas de validación de contraseñas (longitudes, políticas de caducidad), etc.

El problema de la autenticación ya puede ser resuelto mediante el uso de Open ID, pero también existe una necesidad de proporcionar mecanismos de autorización para la explotación de los APIs públicos de servicios como Facebook, Twitter, etc. Es decir, interactuar desde una web en otros servicios, por ejemplo enviar un *tweet* desde una web a un perfil de Twitter o ver el *timeline* de un determinado perfil embebido en otra aplicación [32]. Ahí es donde entra OAuth.

Para detallar el proceso de autenticación y autorización de OAuth introduzcamos en primer lugar algunos conceptos [33].

- Proveedor.

  Plataforma que ofrece el servicio final y que implemente OAuth. Por ejemplo: Twitter.

- Usuario.

  El consumidor del servicio, la persona que va a acceder a los contenidos de un determinado servicio y realizar operaciones sobre él.

- Servidor/Consumidor.

  Aplicación o servicio que se apoya en OAuth para autenticar y autorizar usuarios sobre la plataforma del proveedor. Puede ofrecer sus propios servicios y/o implementar el API del proveedor para ofrecer servicios de este.

Estaríamos hablando de páginas web que en lugar mantener una política propia de autenticación y autorización, se apoyan en OAuth y los servicios de un proveedor. De esta forma ofrecen al usuario una forma de acceso unificada facilitando que los usuarios no tengan que mantener varios perfiles, uno por servicio sino que con el perfil del proveedor pueden acceder a múltiples servicios consumidores.

En la siguiente imagen podemos ver un ejemplo de un portal que en lugar de mantener una gestión propia de perfiles, se apoya en diferentes redes sociales. El usuario de *LiveJournal* no tiene que crearse un perfil específico para este portal sino que puede utilizar su perfil en otra red social para autenticarse y ser autorizado en este servicio.

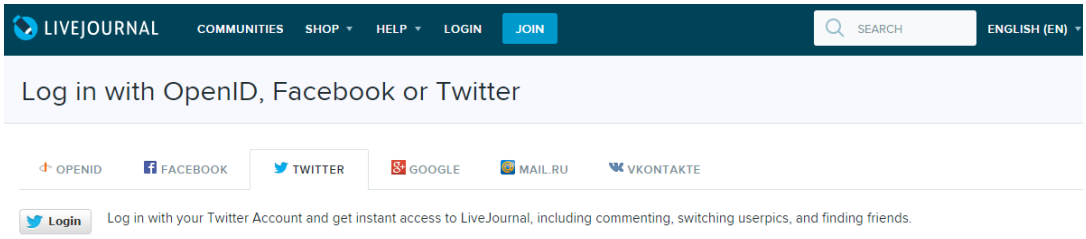

Ilustración 35 – Login en consumidor integrado con proveedor

Desde el punto de vista OAuth, en este ejemplo, *LiveJournal* es un consumidor mientras que Twitter, Facebook o Google+ son proveedores.

- Credenciales OAuth.

  OAuth utiliza tres tipos de credenciales: credenciales de cliente (anteriormente llamadas *consumer key and secret)*, credenciales temporales (anteriormente llamadas *request token and secret*) y credenciales de acceso (anteriormente llamadas *access token and secret*).

  - Credenciales de cliente.
    Autentican al usuario. Son las credenciales que el usuario utilizará siempre que acceda al servicio consumidor.

  - Credenciales de acceso.
    Son las credenciales con las que el servicio consumidor accederá al proveedor de servicios.

    Con el uso de OAuth, el servicio consumidor **no** dispone de los datos de autenticación del usuario. Esto ofrece un grado adicional de protección y confianza en los servicios consumidores dado que finalmente quien autentica es el proveedor y por lo tanto es el único que conoce los datos de acceso del usuario.

    Las credenciales de acceso autorizan al servidor para acceder al proveedor con un conjunto de permisos que le autorizan a realizar un conjunto de operaciones determinado.

Estas credenciales tienen un periodo de caducidad, no están activas indefinidamente. Adicionalmente el usuario puede cancelarlas desde la aplicación del proveedor, dejando sin acceso a los servicios que las utilicen.

- Credenciales temporales.
  Son las credenciales utilizadas por el consumidor en primera instancia, cuando se inicia el proceso de autenticación/autorización con el proveedor de servicios.

Veamos a continuación qué proceso se sigue para autenticar/autorizar un usuario [34].

1. Un usuario va a acceder a un determinado servicio.

Ilustración 36 – OAuth – Solicitud acceso a un servicio

Utilizaremos a modo de ejemplo el portal *LiveJournal.*

2. El servicio consumidor se apoya en un proveedor de servicios para la autenticación y autorización. Desde el punto de vista OAuth, es un consumidor.
   El servicio puede apoyarse en el proveedor simplemente para controlar el acceso de usuarios, sin tener una integración mayor con el proveedor. De esta forma utiliza al proveedor únicamente para proporcionar al usuario un mecanismo sencillo de conexión.
   También puede ser un servicio basado en un determinado proveedor con lo que además de aprovechar el servicio de autenticación, explota activamente el API.

   En este paso, el consumidor obtiene unas credenciales temporales.

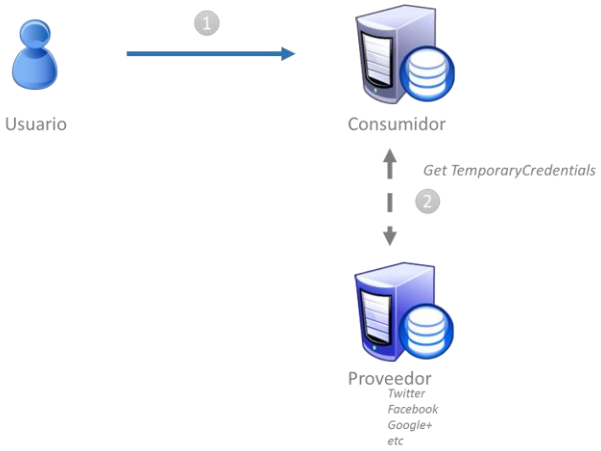

Ilustración 37 – OAuth – Temporary Credentials

3. Con las credenciales temporales, redirige al usuario al proveedor de servicios. Al usuario se le presenta la url OAuth de autenticación del proveedor.

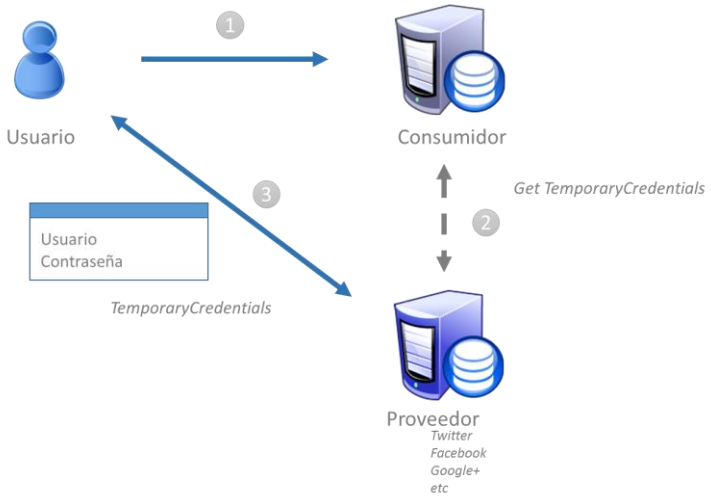

Ilustración 38 – OAuth – Redirección del Consumidor al Proveedor

Siguiendo con el ejemplo del portal *LiveJournal*, seleccionamos el acceso vía Facebook.

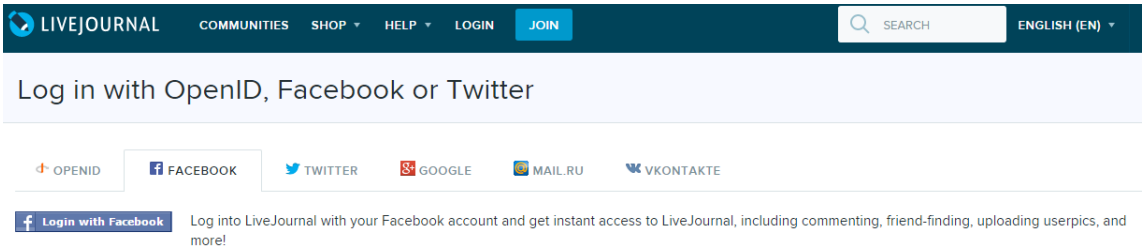

Ilustración 39 – OAuth – Solicitud acceso a un LIVEJOURNAL vía Facebook

Al pinchar *Login with Facebook*, LiveJournal direcciona al usuario a Facebook, de modo que el usuario se conecta mediante su perfil de Facebook. Como puede observarse la validación la realiza Facebook, no LiveJournal.

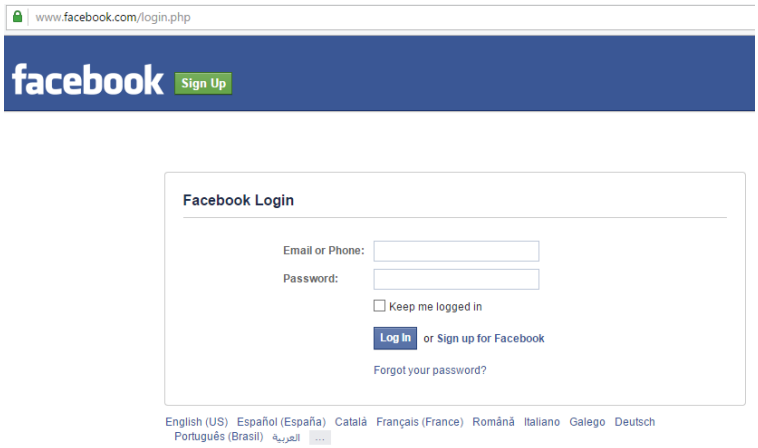

Ilustración 40 – OAuth – Redirección LiveJournal a Facebook

4. El usuario se autentica en la página del proveedor, no se facilitan en ningún momento las credenciales del usuario al servicio consumidor.

   Cuando la validación es correcta, las credenciales temporales se marcan como *client credentials.* Digamos que la autenticación del usuario en el proveedor de servicios ha aceptado esas credenciales.

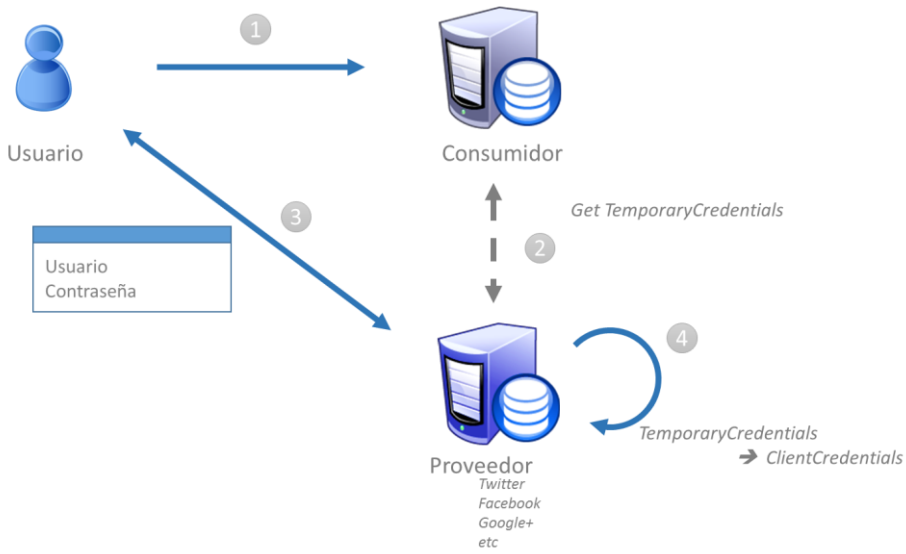

Ilustración 41 – OAuth – Validación de usuario, generación ClientCredentials

La transformación es transparente para el usuario.

En este paso el proveedor pide permiso explícito al usuario para que se permita el acceso del consumidor.

Aquí se aprecia además una de las características de OAuth, diferenciadora de Open Id. La posibilidad de gestionar la no sólo la autenticación del usuario, sino la autorización para que la aplicación consumidora utilice el API del proveedor para realizar operaciones. En este ejemplo, publicar en Facebook en nombre del usuario.

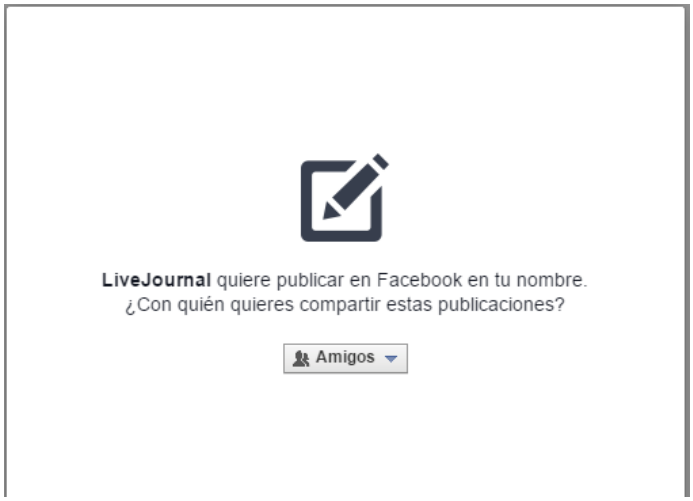

Ilustración 42 – OAuth – Autorización LiveJournal sobre Facebook

5. Las credenciales de cliente viajan hasta el usuario que es redirigido al consumidor. El usuario por lo tanto ya tiene unas credenciales. Está autenticado.

6. La redirección llega al consumidor.

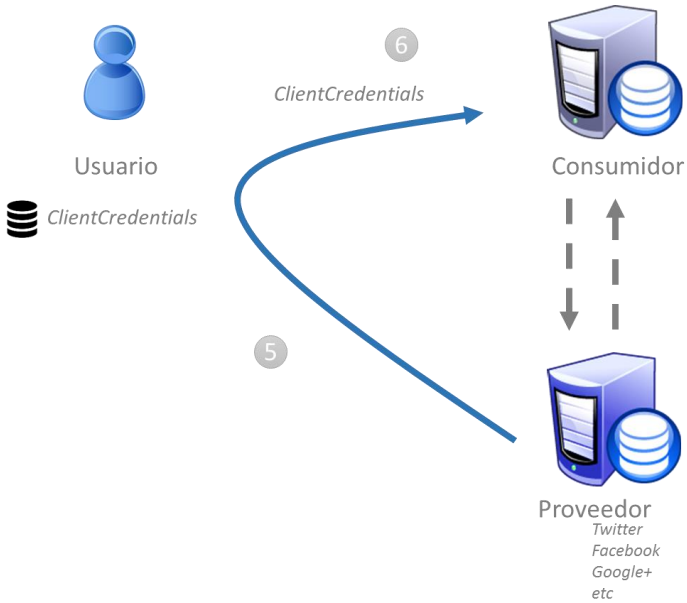

Ilustración 43 – OAuth – Envío de las cliente credentials al usuario y direccionamiento al consumidor

7. A partir de ahí, de forma transparente para el usuario, el consumidor utiliza las credenciales de usuario para negociar el intercambio por unas credenciales de acceso,

   Estas credenciales, en base a las operaciones que haya autorizado el usuario en el paso 4, permitirán al consumidor la realización de unas operaciones u otras.

   Mientras las credenciales de usuario sean válidas, es decir, no hayan caducado o no hayan sido revocadas por el propio usuario, el consumidor podrá realizar tantas operaciones tenga autorizadas mediante la generación de las correspondientes credenciales de acceso.

   En la siguiente captura de pantalla, en la cabecera de *Live Journal* podemos ver cómo se muestran los datos de conexión del usuario mediante su perfil en Facebook.

   También se muestra a modo ejemplo como *Live Journal* posibilita que un usuario utilice para autenticar/autorizar el proveedor de servicios que prefiera, o incluso varios. Nuevamente se hace constar que OAuth no sólo va a autenticar al usuario, sino que va a controlar la autorización del servicio para realizar operaciones en diferentes proveedores.

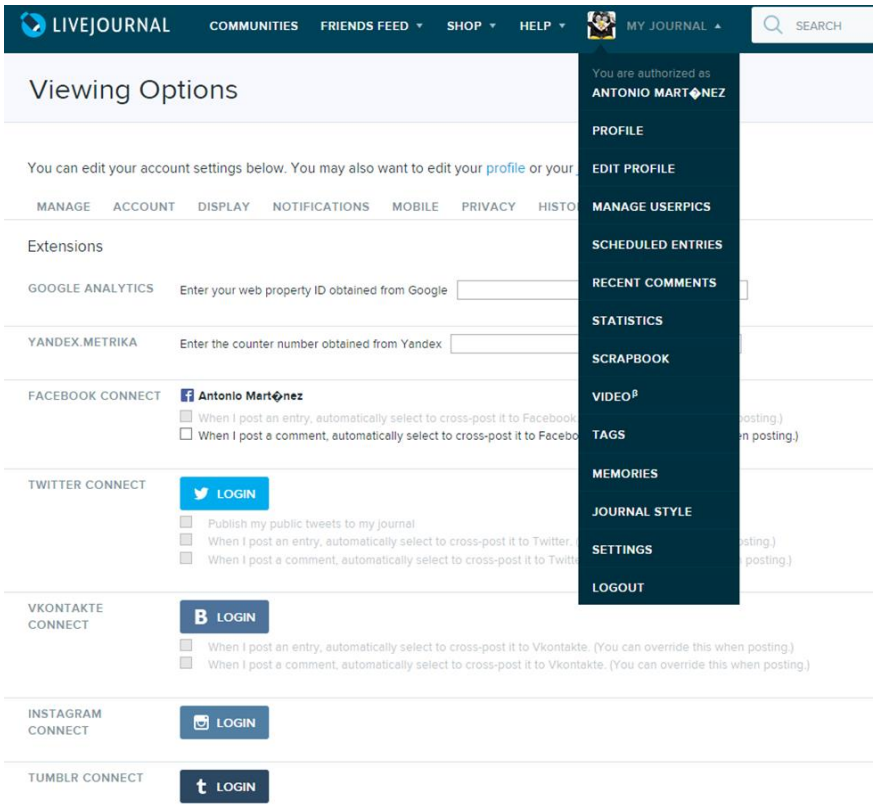

Ilustración 44 – OAuth – Consumidor mostrando credenciales de servicio

3.2.3.3 API REST Twitter en detalle

Una vez introducidas las tecnologías del API Twitter utilizado, entramos en mayor profundidad en el API REST de Twitter.

El API REST de Twitter proporciona un interfaz para realización de operaciones de toda índole. Desde obtener el *timeline* de un usuario, hasta crear listas, buscar *tweets* o enviarlos.

El API necesita que se produzca previamente la autenticación y autorización vía OAuth. Cualquier operación que se realice tendrá que ajustarse a las credenciales de dicho usuario.

El API REST de Twitter dispone de los siguientes objetos:

- Usuarios.

  Representan usuarios del sistema. Un usuario puede crear *tweets*, seguir otros usuarios, crear listas, suscribirse a listas. Puede ser mencionado, seguido, intercambiar mensajes directos, etc.

- *Tweets*.

  Representa a un tweet.
  El objeto contiene todos los elementos que representan al *tweet*, desde el mensaje propiamente dicho hasta el usuario que lo ha enviado, las referencias del mensaje, si se ha utilizado algún *hashtag*, la fecha y hora del mensaje, si es un *retweet*, un mensaje privado o un *tweet* normal, etc.

  Todos los *tweets* tienen un identificador único. Twitter utiliza *snowflake* [35] como tecnología para asegurar la generación de números únicos.

- Entidades.

  Las entidades contienen metadatos con información adicional sobre un contenido publicado en Twitter. La entidad por sí misma no existe, siempre viene dentro de un objeto que la dota de contexto.

  Un ejemplo de entidad serían los *hashtags* de un *tweet*.

  - *Hashtags.*

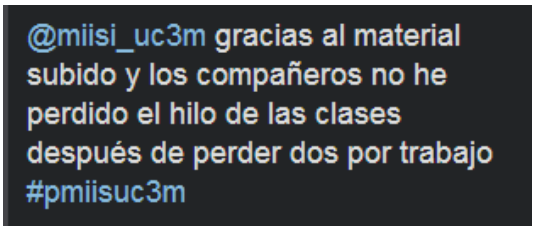

Ilustración 45 – Tweet con hashtag

De esta forma queda representada la entidad *hashtags.*

```
"hashtags":  [
    {
            "text": "pmiisuc3m",
            "indices":  [37, 47]
        }
    ]
```

Otros ejemplos de entidades pueden ser las fotos, las urls incluidas en el *tweet* o las menciones que aparecen en un *tweet.*

- Places.

Representa una localización específica representada por sus coordenadas.
La localización de un usuario o un *tweet* es especialmente relevante para la plataforma a la hora de ofrecer contenidos en base a la localización. Por ejemplo para sugerir los *treding topic* locales al usuario.

Basado en los objetos anteriores y en los elementos generales de configuración, las siguientes tablas muestran todas las operaciones disponibles [36].

- Operaciones relacionadas con el *timeline*

Operación	**Petición HTTP**
Devuelve las menciones del usuario.	GET/statuses/mentions_timeline.json
Tweets más recientes posteados por el usuario.	GET/statuses/user_timeline.json
Colección de los *tweets y retweets* realizados por el usuario así como de los usuarios que se sigue.	GET/statuses/home_timeline.json

Tabla 8 – Twitter API REST - timeline

- Operaciones relacionadas con *tweets*

Operación	**Petición HTTP**
Colección de los 100 *retweets* más recientes del *tweet* pasado como parámetro.	GET/statuses/retweets/{id}.json
Retorna el *tweet* pasado como parámetro.	GET/statuses/show/{id}.json
Elimina el *tweet* indicado. Sólo puede hacerlo el autor del mismo.	POST/statuses/destroy/{id}.json
Actualiza el estado del usuario.	POST/statuses/update.json
Hace un *retweet* del *tweet* indicado	POST/statuses/retweet/{id}.json
Devuelve el *tweet* solicitado en formato Embeb.	GET/statuses/oembed.json
Retorna la colección de los 100 usuarios que más recientemente han hecho un *retweet* del *tweet* indicado.	GET/statuses/retweets/{id}.json

Tabla 9 – Twitter API REST - tweet

- Operaciones relacionadas con búsquedas de *tweets*

Operación	Petición HTTP
Devuelve los *tweets* que se ajusten a la consulta pasada.	GET /search/tweets.json

Tabla 10 – Twitter API REST - Búsquedas

- Operaciones relacionadas con ayuda

Operación	Petición HTTP
Devuelve datos de la configuración de Twitter como las máximas resoluciones de las fotos así como la longitud de las Urls que se pueden adjuntar, etc.	GET/help/configuration.json
Lista de lenguajes soportados por Twitter.	GET/help/languages.json
Devuelve la política de privacidad que aplica en ese momento.	GET/help/privacy.json
Acuerdo de servicio que aplica en ese momento.	GET/help/tos.json
Dado que el número de peticiones que se pueden realizar con el API es limitado, esta operación informa del estado del usuario de cara a poder o no realizar más peticiones.	GET/application/rate_limit_status.json

Tabla 11 – Twitter API REST - Ayuda

- Operaciones relacionadas con la notificación de *SPAM*

Operación	Petición HTTP
Permite denunciar ante Twitter que una cuenta es SPAM.	POST /users/report_spam.json

Tabla 12 – Twitter API REST - SPAM

- Operaciones relacionadas con la *trends*

Operación	Petición HTTP
Devuelve los *treding topics* de una determinada localización.	GET/trends/place.json
Retorna un conjunto de localizaciones en las que se dispone de información sobre *treding topics*. Todas de las que se tenga información.	GET/trends/available.json
Devuelve los *treding topics* de localizaciones cercanas a la facilitada	GET/trends/closest.json

Tabla 13 – Twitter API REST – Trends

- Operaciones relacionadas con geolocalización

Operación	Petición HTTP
Devuelve toda la información disponible de un determinado lugar.	GET/geo/id/{place_id}.json
Dada una latitud y longitud, se buscan los 20 lugares que pueden enviarse a Twitter en el caso de que el usuario actualice su perfil.	GET/geo/reverse_geocode.json
Busca los lugares asociados al usuario en base a su latitud/longitud e incluso su IP.	GET/geo/search.json

Tabla 14 – Twitter API REST – Geolocalización

- Operaciones relacionadas con búsquedas almacenadas

Operación	Petición HTTP
Lista de búsquedas almacenadas por el usuario.	GET/saved_searches/list.json
Información de la lista facilitada como parámetro.	GET/saved_searches/show/{id}.json
Crea una nueva lista.	POST/saved_searches/create.json
Elimina una lista existente.	POST/saved_searches/destroy/{id}.json

Tabla 15 – Twitter API REST – Búsquedas almacenadas

- Operaciones relacionadas con favoritos

Operación	Petición HTTP
Lista los *tweets* marcados como favoritos.	GET/favorites/list.json
Marca un *tweet* como favorito.	POST/favorites/create.json
Elimina la marca de favorito.	POST/favorites/destroy.json

Tabla 16 – Twitter API REST – Favoritos

- Operaciones relacionadas con sugerencias de usuarios a los que seguir

Operación	Petición HTTP
Lista de los usuarios en una determinada categoría.	GET/users/suggestions/{slug}.json
Lista de sugerencias a las que seguir.	GET/users/suggestions.json
Devuelve los usuarios de una determinada categoría con su estado.	GET/users/suggestions/{slug}/members.json

Tabla 17 – Twitter API REST – Sugerencias usuarios

- Operaciones relacionadas con mensajes directos

Operación	Petición HTTP
Listado de los últimos 20 mensajes privados enviados.	GET/direct_messages.json
Enviar un mensaje privado.	GET/direct_messages/sent.json
Mostrar un mensaje privado.	GET/direct_messages/show.json
Crear un mensaje privado.	POST/direct_messages/new.json

Operación	Petición HTTP
Elimina un mensaje privado.	POST/direct_messages/destroy.json

Tabla 18 – Twitter API REST – Mensajes

- Operaciones relacionadas con amigos y followers.

Operación	Petición HTTP
Obtiene una colección de los usuarios a los que se está siguiendo.	GET/friends/ids.json
Listado de los usuarios que están siguiendo.	GET/followers/ids.json
Muestra la relación del usuario autenticado con los usuarios que se le pasen.	GET/friendships/lookup.json
Listado de las personas que han solicitado ser seguidas pero sobre las que el usuario autenticado todavía no ha aceptado.	GET/friendships/incoming.json
Colección de identificadores de cada usuario que tiene una petición pendiente de aceptación por parte del usuario autenticado.	GET/friendships/outgoing.json
Crea una petición de seguimiento.	POST/friendships/create.json
Elimina una petición de seguimiento.	POST/friendships/destroy.json
Permite habilitar y deshabilitar *retweets* para el pasado como parámetro.	POST/friendships/update.json
Muestra la relación entre los dos usuario que se pasan como parámetro	GET/friendships/show.json

Tabla 19 – Twitter API REST – Amigos y followers

- Operaciones relacionadas con usuarios

Operación	Petición HTTP
Devuelve los parámetros de configuración del usuario autenticado.	GET/account/settings.json
Operación de actualización de la configuración del usuario autenticado.	POST/account/settings.json
Habilita o deshabilita la posibilidad de que Twitter envíe las actualizaciones al usuario autenticado por SMS.	POST/account/update_delivery_device.json
Permite la actualización de algunos parámetros del usuario: nombre, url asociada al perfil, localización, descripción, etc.	POST/account/update_profile.json
Similar al anterior, pero en este caso se centra en la imagen de fondo del perfil, se puede cambiar o deshabilitar.	POST/account/update_profile_background_image.json
En este caso se entra en la posibilidad de cambiar los colores del perfil.	POST/account/update_profile_colors.json
Este método hace habilita el cambio de imagen del perfil.	POST/account/update_profile_image.json
Devuelve los usuarios que están bloqueados por el perfil autenticado.	GET/blocks/list.json

Operación	Petición HTTP
Igual que el anterior pero en lugar de retornar un array de objetos de usuario, retorna un array con los identificadores de los usuarios bloqueados.	GET/blocks/ids.json
Permite bloquear un usuario.	POST/blocks/create.json
Permite eliminar el bloqueo a un usuario.	POST/blocks/destroy.json
Devuelve el objeto usuario completo del usuario o usuarios que se esté consultando. El objeto usuario contiene todos los datos públicos del usuario como su nombre, localización, número de usuarios a los que sigue, último *tweet*, etc.	GET/users/lookup.json
Igual que el anterior pero sólo permite la consulta de un perfil.	GET/users/show.json
Igual que los anteriores en lo referente al resultado solo que este método permite la consulta por un criterio diferente al nombre o identificador de usuario. Se puede componer una query de búsqueda	GET/users/search.json

Tabla 20 – Twitter API REST – Usuarios

- Operaciones relacionadas con listas.

  Existen dos tipos de relaciones de un perfil con las listas.

  - "Miembro de" Los miembros de una lista son los perfiles que se han añadido a la lista. Hay que tener en cuenta que una lista no es más que una forma de ordenar/agrupar perfiles a los que se sigue.

  - "Suscrito a" Las listas pueden hacerse públicas de forma que otros usuarios pueden hacerse followers de la lista, es decir, suscribirse y seguirlas.

  Existen métodos en el API para manejar ambos tipos de membresía.

Operación	Petición HTTP
Retorna todas las listas a las que está subscrito el usuario que autenticado o que se pase por parámetro.	GET/lists/list.json
Devuelve los *tweets* de la lista.	GET/lists/statuses.json
Saca un miembro de la lista. El usuario ha de ser el propietario de la lista.	POST/lists/members/destroy.json
Consulta si un usuario está en una lista.	GET/lists/memberships.json
Devuelve los usuarios suscritos a la lista. Si quien lanza el comando es el propietario de la lista también retorna los privados.	GET/lists/subscribers.json
Para suscribirse a una lista.	POST/lists/subscribers/create.json
Consulta si el usuario está en una lista. Retorna los datos del usuario.	GET/lists/subscribers/show.json

Operación	Petición HTTP
Saca al usuario autenticado de la lista.	POST/lists/subscribers/destroy.json
Añade usuarios a la lista.	POST/lists/members/create_all.json
Consulta la pertenencia a una lista de un conjunto de usuarios.	GET/lists/members/show.json
Lista los miembros que componen una lista.	GET/lists/members.json
Añade un miembro a una lista.	POST/lists/members/create.json
Elimina una lista.	POST/lists/destroy.json
Permite cambiar los atributos de la lista. Nombre, si es pública o privada, etc.	POST/lists/update.json
Crea una lista.	POST/lists/create.json
Retorna los atributos de la lista. Nombre, si es pública o privada, etc.	GET/lists/show.json
Retorna las listas suscritas por un usuario.	GET/lists/subscriptions.json
Elimina todos los miembros de una lista.	POST/lists/members/destroy_all.json

Tabla 21 – Twitter API REST – Listas

En la página https://dev.twitter.com puede consultarse con mayor detalle y profundidad la descripción del propio API, documentación, aplicaciones de ejemplo y foros de discusión.

Antes de utilizar el API es recomendable repasar los acuerdos y restricciones de servicio que Twitter establece [37].

3.2.4 Proceso de Recolección

Llegados al diseño del proceso de recolección nos encontramos con tres posibilidades.

- Utilización de herramientas de terceros para la obtención de los *tweets*.

  Dado que tanto los APIs anteriormente descritos como los estándares sobre los que se sustentan están bien documentados en la propia página de Twitter [38]. Y teniendo además en cuenta los datos de volumetría mostrados en el estado de la cuestión al respecto de Twitter, no es difícil encontrar herramientas que mediante la utilización de los APIs, se conecten a Twitter y recuperen la información de un determinado perfil.

  Herramientas que por otro lado están mayormente pensadas para dos tipos de funcionalidades:

  - Como cliente Twitter.
    En sustitución del cliente web (o cliente pesado para *smartphones*) que proporciona el propio Twitter, existen herramientas para gestionar uno o varios perfiles Twitter. Aquí se encontrarían herramientas como la ya mencionada Tweet Deck [19]

  - Para gestión de perfiles en representación de empresas o productos.
    Estaríamos hablando de herramientas para el uso de los administradores de perfiles corporativos (*community managers)*. Dichos perfiles tienen unas demandas más relacionadas con las estadísticas de uso, el éxito de una determinada publicación, el seguimiento de una marca o producto en redes sociales, etc. [39]

Los requisitos del proyecto en el sentido de la recolección no quedan realmente cubiertos de una forma clara por estas herramientas que al tener un ámbito más comercial no ofrecen una opción tan directa como la descarga de todos los *tweets* de un perfil.

- Desarrollo de una herramienta dentro del ámbito del proyecto para la recolección de los datos.

  Dentro del ámbito del proyecto, estamos siempre hablando como requisitos de recolección de la posibilidad de realizar una extracción a demanda en un determinado momento del proyecto, al finalizar el curso. El desarrollo de una herramienta implementando las tecnologías descritas para una única extracción, resulta costoso más teniendo en cuenta que existen otras alternativas.

  Si es cierto que es un trabajo que considero puede ser de utilidad y que de ahí que se haya incluido en el capítulo 7.3 Otros Trabajos.

- Utilización de herramientas ofrecidas por Twitter.

  Esta ha sido la opción escogida.

  Dentro de la página de desarrolladores, Twitter ofrece algunas utilidades a modo de muestra de las funcionalidades de su API.
  En concreto existe una herramienta, API Console [31] que explota el API REST y cuyo uso cumple con los requisitos marcados.
  En la parte de implementación se muestra su uso y su aplicación al proyecto.

3.3 Transformación / Preparación

Una vez dispuestos los *tweets,* el siguiente paso del proceso de KDD consiste en el tratamiento de la información en bruto para refinarla y darle uniformidad. Todo ello con el objetivo de preparar los datos para la posterior tarea de minería de datos.

El trabajo de minería de datos seguido en el presente trabajo se basa en la utilización de la herramienta *knowledgeMANAGER* [40], herramienta que permite la definición de ontologías, representación de expresiones mediante la creación de patrones y definición de relaciones entre ellos.

Una característica de la configuración de la herramienta para su utilización en este trabajo es que viene precargada con una gramática de lenguaje natural en castellano.

Como se ha visto con anterioridad los datos de Twitter obtenidos mediante la explotación de su API REST son devueltos en formato json por lo que no son directamente operables mediante una gramática de lenguaje natural. Si además le añadimos que en cada *tweet* el número de propiedades que se retorna es muy elevado y la mayoría de ellas no son relevantes para el estudio, el proceso de transformación y preparación de los datos se hace imprescindible.

Para la consecución de esta tarea se ha elegido la opción de desarrollar una utilidad *transformTwitterData* que dado un fichero de texto con el contenido en formato json de n *tweets* (tal y como hayan sido devueltos por el API REST de Twitter) es capaz de serializar cada *tweet* en una única línea con la siguiente información:

Tweet ID. Identificador único del *tweet*.
Usuario. Identificador del usuario que ha escrito el *tweet*.
Identificador del *tweet* al que responde. En el caso de que el *tweet* sea una respuesta a otro *tweet*, se incluye el *tweet ID* al que se responde.
Texto del *tweet*. Contenido del *tweet*.

Adicionalmente y con el objetivo de hacer aún más legible la información, incluye texto de lenguaje natural que permite la lectura natural de la información objeto del estudio.

A modo de ejemplo se muestra la conversión que realiza la utilidad sobre un *tweet* en formato json.

- *Tweet* tras el proceso de conversión.

  "En el tweet: 605076621717053400, en respuesta al tweet: 573776088377262100, el usuario roxana10720373 comenta: miisiuc3m A mí me gusta más el tema de estrategia y transición de servicio, ya que son el punto inicial e intermedio de todo el ciclo."

- Objeto *tweet* en formato json.


```
{
"created_at": "Sun May 31 18:21:23 +0000 2015",
"id": 605076621717053400,
"id_str": "605076621717053440",
"text": "@miisi_uc3m A mi me gusta más el tema de estrategia y transición de servicio, ya que son el punto inicial e
intermedio de todo el ciclo.",
"source": "<a href="http://twitter.com" rel="nofollow">Twitter Web Client</a>",
"truncated": false,
"in_reply_to_status_id": 573776088377262100,
"in_reply_to_status_id_str": "573776088377262080",
"in_reply_to_user_id": 3010797699,
"in_reply_to_user_id_str": "3010797699",
"in_reply_to_screen_name": "miisi_uc3m",
"user": {
"id": 2455299134,
"id_str": "2455299134",
"name": "roxana",
"screen_name": "roxana10720373",
"location": "",
"description": "",
"url": null,
"entities": {
"description": {
"urls": []
}
},
"protected": false,
"followers_count": 7,
"friends_count": 13,
"listed_count": 0,
"created_at": "Sun Apr 20 17:04:26 +0000 2014",
"favourites_count": 1,
"utc_offset": null,
"time_zone": null,
"geo_enabled": false,
"verified": false,
"statuses_count": 2,
"lang": "es",
```

```
"contributors_enabled": false,
"is_translator": false,
"is_translation_enabled": false,
"profile_background_color": "C0DEED",
"profile_background_image_url": "http://abs.twimg.com/images/themes/theme1/bg.png",
"profile_background_image_url_https": "https://abs.twimg.com/images/themes/theme1/bg.png",
"profile_background_tile": false,
"profile_image_url": "http://pbs.twimg.com/profile_images/605073705262292993/zIAob8j7_normal.jpg",
"profile_image_url_https": "https://pbs.twimg.com/profile_images/605073705262292993/zIAob8j7_normal.jpg",
"profile_banner_url": "https://pbs.twimg.com/profile_banners/2455299134/1433096023",
"profile_link_color": "0084B4",
"profile_sidebar_border_color": "C0DEED",
"profile_sidebar_fill_color": "DDEEF6",
"profile_text_color": "333333",
"profile_use_background_image": true,
"has_extended_profile": false,
"default_profile": true,
"default_profile_image": false,
"following": true,
"follow_request_sent": false,
"notifications": false
},
"geo": null,
"coordinates": null,
"place": null,
"contributors": null,
"is_quote_status": false,
"retweet_count": 0,
"favorite_count": 0,
"entities": {
"hashtags": [],
"symbols": [],
"user_mentions": [
{
"screen_name": "miisi_uc3m",
"name": "MIISI",
"id": 3010797699,
"id_str": "3010797699",
"indices": [
0,
11
]
}
],
"urls": []
},
"favorited": false,
"retweeted": false,
"lang": "es"

},
```


La siguiente ilustración muestra gráficamente cómo se enlazan los procesos de Recolección y Transformación.

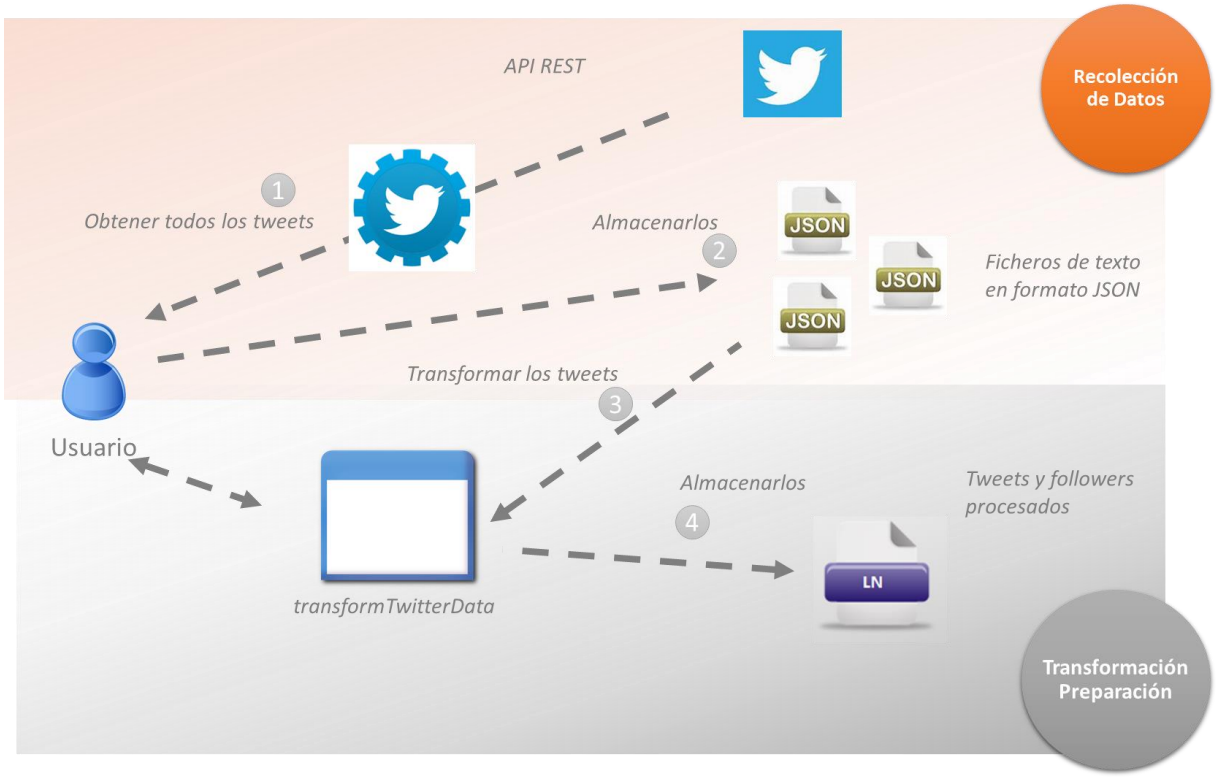

Ilustración 46 –Flujo Recolección y Transformación

1. Obtención de los *tweets* y *followers* mediante el API REST de Twitter.

2. Almacenamiento en ficheros con el formato original json.

Una vez que se dispone de los datos y mediante la utilización de la aplicación desarrollada *transformTwitterData:*

3. Procesamiento y transformación de datos a información

4. Almacenamiento del resultado del procesamiento.

3.3.1 TransformTwitterData

Entramos a continuación a detallar las características de implementación y el funcionamiento de la utilidad *transformTweetsData.*

3.3.1.1 Requisitos Funcionales

La utilidad se desarrolla con un único objetivo; convertir la información devuelta por Twitter en bruto a un formato legible y tratable por knowledgeMANAGER.

Se detectan los siguientes Requisitos Funcionales.

Requisito	RF-001
Nombre	Procesar *tweets*
Descripción	Dado un fichero con *tweets* en formato json, se procesan y se serializan en una línea que lo represente. Se incluyen los siguientes datos: *Tweet ID*. Identificador único del *tweet*. Usuario. Identificador del usuario que ha escrito el *tweet*. Identificador del *tweet* al que responde. En el caso de que el *tweet* sea una respuesta a otro *tweet*, se incluye el *tweet ID* al que se responde. Texto del *tweet*. Contenido del *tweet*. La línea de salida tendrá el siguiente formato. *"En el tweet: <tweetID>, en respuesta al tweet: <tweetID>, el usuario <usuario> comenta: <texto del tweet>"*
Necesidad	Alta
Dependencias	Se trabajará sobre los ficheros obtenidos en el proceso de Recolección de Datos.

Tabla 22 – transformTwitterData RF001 – Procesar Tweets

Requisito	RF-002
Nombre	Exportar *tweets*
Descripción	El resultado del tratamiento de *tweets* ha de ser exportado a un fichero en formato texto. Cada línea del fichero, representa un *tweet* serializado.
Necesidad	Alta
Dependencias	RF001. Se exportan los *tweets* serializados obtenidos tras el procesamiento en RF01.

Tabla 23 – transformTwitterData RF002 – Exportar tweet

Requisito	RF-003
Nombre	Limpiar caracteres
Descripción	Para evitar problemas en el procesamiento con *knowledgeMANAGER* se eliminarán caracteres problemáticos: Smiles (☺ ☹, etc.). _ @ etc.
Necesidad	Alta
Dependencias	RF001 El tratamiento se ha de hacer el propio proceso de conversión para evitar propagar estos caracteres.

Tabla 24 – transformTwitterData RF003 – Exportar tweet tratado

3.3.1.2 Requisitos No Funcionales

- Requisitos de Arquitectura.

No se han definido requisitos relativos a la arquitectura de la aplicación, motivado porque se ha desarrollado no como objetivo final del proyecto sino como elemento accesorio para facilitar la labor de transformación.

Se ha utilizado una arquitectura de desarrollo basada en elementos estándar de mercado.

- IDE de desarrollo: Microsoft Visual Studio 2010.
  Como alternativa libre y multiplataforma se puede utilizar *sharpdevelop* [41]

- Framework.
  Microsoft .NET Framework 4.0.
  Como alternativa libre para plataformas Linux se puede utilizar *mono* [42]

- Requisitos de Interfaz Gráfico

Requisito	**RIGU-001**
Nombre	Procesar Menciones de usuario
Descripción	Entrada específica para procesar un fichero con los *tweets* de menciones del usuario. Tal y como se indica más adelante, en el apartado 5.1 Recolección de Datos, unos de los *tweets* objetivo del estudio son los *tweets* en los que se menciona al usuario son objeto del estudio. Entre este tipo de *tweets* se encuentra la mayor parte del contenido.
Necesidad	Alta
Dependencias	Sin dependencias

Tabla 25 – transformTwitterData RIGU001 – Menciones

Requisito	**RIGU-002**
Nombre	Procesar Mensajes de usuario
Descripción	Entrada específica para procesar un fichero con los *tweets* que ha escrito el usuario @miisi_uc3m. El usuario de la asignatura. Tal y como se indica más adelante, en el apartado 5.1 Recolección de Datos estos *tweets* también son objeto del estudio ya que contienen los *tweets*, que se han escrito desde el equipo docente y existen algunos que no se encuentran dentro del listado de menciones ya que no existe la necesidad de que en el *tweet* se mencione a la propia cuenta de la asignatura.
Necesidad	Alta
Dependencias	Sin dependencias

Tabla 26 – transformTwitterData RIGU002 – Tweets @miisi_uc3m

Requisito	**RIGU-003**
Nombre	Obtener *followers*
Descripción	Dado que se cargarán en *knowledgeMANAGER* los usuarios que han participado enviando *tweets* y con el objeto de tener el listado abstraído del conjunto global de *tweets* se ve conveniente la posibilidad de obtener todos los usuarios que siguen la cuenta de la asignatura.
Necesidad	Media

Requisito	RIGU-003
Dependencias	Sin dependencias

Tabla 27 – transformTwitterData RIGU003 – Followers

3.3.1.3 Casos de Uso

La aplicación cubre los siguientes casos de uso.

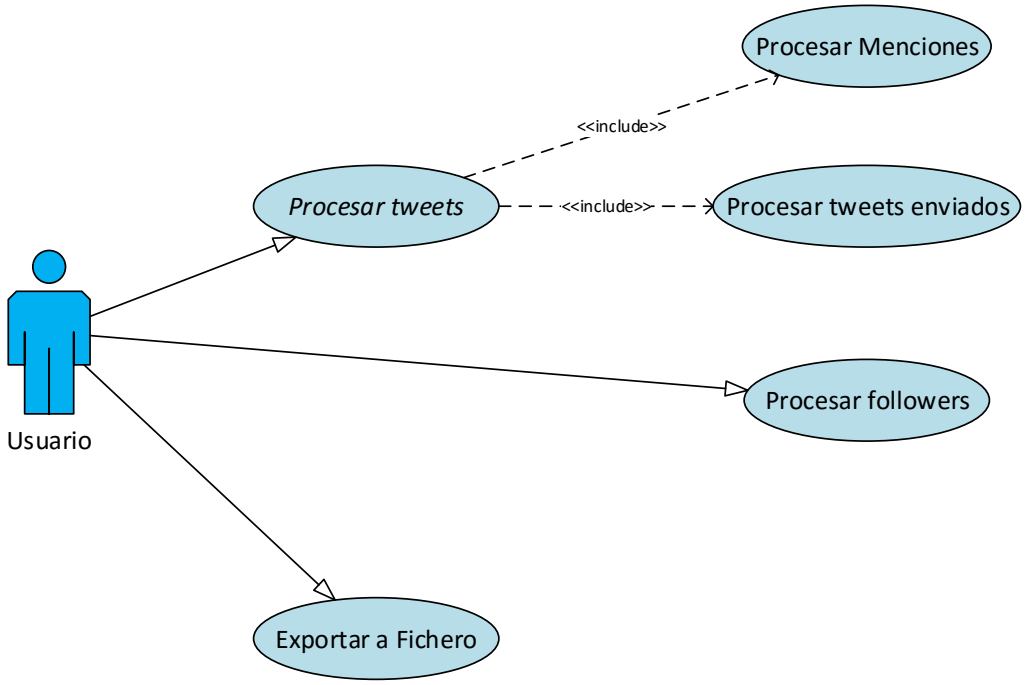

Ilustración 47 –Casos de Uso

Caso de uso	CU-001
Nombre	Procesar tweets
Actores	Usuario
Objetivo	Procesar un fichero con *tweets* en formato json y serializarlo en texto plano
Descripción	Se serializan los *tweets* de forma que cada *tweet* es representado por una única línea con los datos relevantes del mismo. *Tweet ID*. Identificador único del *tweet*. Usuario. Identificador del usuario que ha escrito el *tweet*. Identificador del *tweet* al que responde. En el caso de que el *tweet* sea una respuesta a otro *tweet*, se incluye el *tweet ID* al que se responde. Texto del *tweet*. Contenido del *tweet*.
Pre-condición	Se debe disponer del fichero con los *tweets* en formato json tal y como lo genera el API REST de Twitter.
Post-condición	Los *tweets* en formato: *"En el tweet: <tweetID>, en respuesta al tweet: <tweetID>, el usuario <usuario> comenta: <texto del tweet>*

Tabla 28 – Caso de Uso CU-001 - Procesar tweets

Caso de uso	CU-002

Nombre	Procesar *followers*
Actores	Usuario
Objetivo	Procesar un fichero con todos los *followers* de la cuenta de la asignatura (@miisi_uc3m) en formato json y extraer únicamente el nombre del usuario.
Descripción	Se extraen los nombres del usuario.
Pre-condición	Se debe disponer del fichero con los *followers* en formato json tal y como lo genera el API REST de Twitter.
Post-condición	Nombre del usuario.

Tabla 29 – Caso de Uso CU-002 - Procesar followers

Caso de uso	**CU-003**
Nombre	Exportar a Fichero
Actores	Usuario
Objetivo	Exportar a un fichero de texto el resultado del procesamiento de *tweets* o *followers*.
Descripción	Las operaciones anteriores muestran los datos del procesamiento. Este caso de uso cubre la necesidad de exportar esos datos a un fichero plano una vez validados.
Pre-condición	Han de existir datos procesados, sino se generaría un fichero vacío. También ha de indicarse la ruta y nombre del fichero de salida.
Post-condición	Datos procesados en el fichero indicado.

Tabla 30 – Caso de Uso CU-003 – Exportar a Fichero

Caso de uso	**CU-004**
Nombre	Procesar Menciones
Actores	Usuario
Objetivo	Exportar a un fichero de texto el resultado del procesamiento de *tweets* o *followers*.
Descripción	Se serializan los *tweets* de forma que cada *tweet* es representado por una única línea con los datos relevantes del mismo. *Tweet ID*. Identificador único del *tweet*. Usuario. Identificador del usuario que ha escrito el *tweet*. Identificador del *tweet* al que responde. En el caso de que el *tweet* sea una respuesta a otro *tweet*, se incluye el *tweet ID* al que se responde. Texto del *tweet*. Contenido del *tweet*.
Pre-condición	Fichero con las menciones según se extrae del uso del API REST de Twitter.
Post-condición	Menciones en formato: *"En el tweet: <tweetID>, en respuesta al tweet: <tweetID>, el usuario <usuario> comenta: <texto del tweet>*

Tabla 31 – Caso de Uso CU-004 – Procesar Menciones

Caso de uso	**CU-005**
Nombre	Procesar *tweets* de usuario
Actores	Usuario
Objetivo	Exportar a un fichero de texto el resultado del procesamiento de *tweets* del usuario de la cuenta teniendo en cuenta no repetir aquellos no relevantes (re-tweets) y los que aparecen en Menciones.

Caso de uso	CU-005
Descripción	Se serializan los *tweets* de forma que cada *tweet* es representado por una única línea con los datos relevantes del mismo. *Tweet ID*. Identificador único del *tweet*. Usuario. Identificador del usuario que ha escrito el *tweet*. Identificador del *tweet* al que responde. En el caso de que el *tweet* sea una respuesta a otro *tweet*, se incluye el *tweet ID* al que se responde. Texto del *tweet*. Contenido del *tweet*. Este caso de uso es muy similar al anterior pero hay que tener en cuenta que su objetivo es extraer aquellos *tweets* relevantes para el estudio que han quedado fuera del listado de Menciones, es decir, sólo sacará aquellos *tweets* enviados por el usuario *miisi_uc3m* que: NO sean *retweets*. NO sean menciones sobre sí mismo (ya incluidos en CU-004).
Pre-condición	Fichero con los tweets del usuario según se extrae del uso del API REST de Twitter.
Post-condición	*Tweets* en formato: *"En el tweet: <tweetID>, en respuesta al tweet: <tweetID>, el usuario <usuario> comenta: <texto del tweet>*

Tabla 32 – Caso de Uso CU-005 – Procesar tweets de usuario

3.4 Minería de sentimientos

Una vez extraída la información y tratada para uniformar y aislar los conjuntos de información objeto del análisis, entramos en el proceso de minería de datos.

Para el desarrollo del presente trabajo se ha optado por realizar un desarrollo del proceso de minería de datos asistido por ontologías. Esto es, a partir de la información disponible se desarrolla una ontología en la que se conceptualiza el universo de *tweets* de la asignatura objeto del estudio y se utiliza la ontología para representar los resultados.

El modelo se construirá a partir de la identificación de los diferentes conceptos o elementos que componen un *tweet*, composición de los patrones sintácticos del *tweet* y composición de las relaciones y meta-propiedades que proporcionan la semántica del *tweet*: *tweets* positivos, de opinión, negativos, sobre estándares, etc.

A partir de aquí se trabajará en la pertenencia de un *tweet* a un patrón determinado u otro y el análisis de las relaciones entre patrones.

Para el desarrollo de la ontología se ha utilizado la herramienta knowledgeMANAGER de la compañía The REUSE Company [40]. La herramienta está diseñada y desarrollada para el trabajo con ontologías. Tiene por lo tanto herramientas para la creación de los diferentes elementos: términos, tokens, patrones, clústeres, etc., así como herramientas para la definición y trabajo con patrones. Es además la herramienta utilizada por el departamento KNOWLEDGE REUSE GROUP [43], departamento en el que se desarrolla el presente proyecto.

Previo a la implementación de en la herramienta, en la fase de análisis se ha producido una catalogación de los *tweets* en varios conceptos. Estos conceptos son los que se aplicarán a la hora de implementar la ontología e indexar e interpretar los *tweets*.

- Tweets positivos.

  Tweets que se expresan en un carácter marcadamente positivo: A mí me gusta, me ha encantado, muy interesante, etc.

- Tweets negativos.

  En contraposición a los anteriores, *tweets* en los que se expresa alguna cuestión negativa.

- Tweets de opinión.

  Expresan una opinión: En mi opinión, creo que, me parece, etc.

- Tweets de pregunta.

  El usuario realiza alguna pregunta: ¿qué opináis?, ¿Por qué este en particular?, etc.

- Estándares o certificaciones.

  Son *tweets* en los que aparecen nombrados estándares o certificaciones: ITIL, CISA, COBIT, etc.

- Sarcasmo.

  El sarcasmo es una cuestión más semántica que gramatical, para el reconocimiento de este tipo de *tweets* nos hemos apoyado en el análisis del *hashtag* definido *#smiisuc3m*.

3.5 Entorno Tecnológico

El proyecto se ejecuta interactuando con diferentes actores, cada una de las labores de la minería exige la interactuación con un sistema que tiene su propia arquitectura y reglas de acceso e integración.

La primera fase de extracción de la información se realiza directamente sobre Twitter. Twitter ofrece un API abierto y un conjunto de aplicaciones de ejemplo que lo explotan. Se ha utilizado una consola accesible desde la propia web de desarrolladores de Twitter.

La consola implementa el API REST con sistema de autenticación/autorización OAuth y permite descargar cualquier objeto de Twitter (*tweets*, usuarios, *followers*, etc.).

La información obtenida directamente de Twitter está formateada mediante json. El formato es claro y preciso en la definición de los diferentes objetos Twitter, pero no es apto para su procesamiento como lenguaje natural. Es por ello que se hace necesaria la utilización de una herramienta de transformación de los *tweets* a lenguaje natural.

Para la transformación se ha realizado un desarrollo dentro del proyecto. Se ha desarrollado una utilidad con tecnología Microsoft .NET Framework que dado un objeto Twitter lo convierte en lenguaje natural.

Finalmente los *tweets* convertidos a lenguaje natural son procesados por la herramienta knowledgeMANAGER, herramienta especializada en la definición y explotación de ontologías. Es con esta herramienta con la que se completa el proceso de minería de sentimientos.

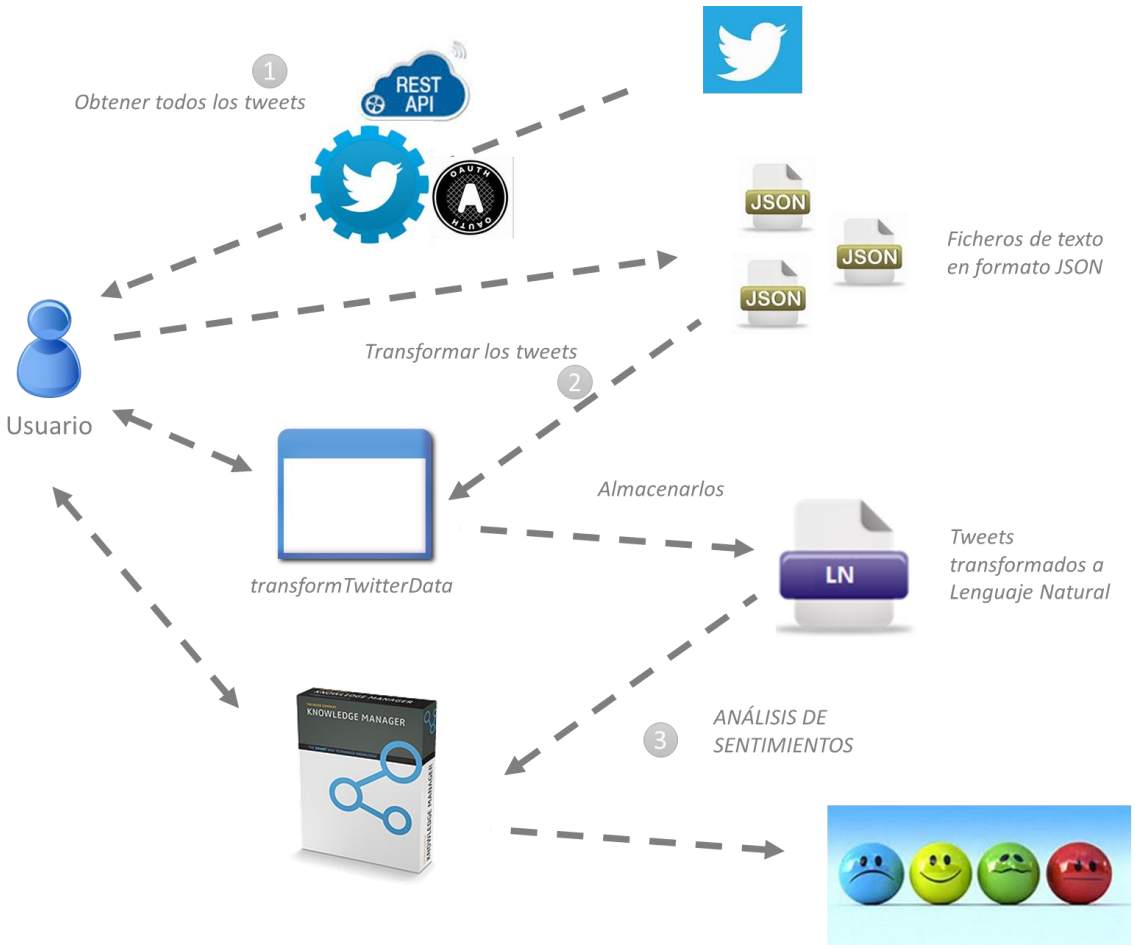

Ilustración 48 – Entorno Tecnológico

Capítulo 4

Planificación y Presupuesto

En este capítulo se detallan las diferentes fases del proyecto, los recursos asignados y los costes totales del proyecto, incluyendo costes personales y costes materiales.

4. Planificación y Presupuesto

En este apartado se muestra un diagrama de Gantt con la planificación del proyecto. Se desarrollan todas y cada una de las tareas que se han ejecutado así como la valoración de tiempo asignada.

También se desarrolla el mapa de costes global del proyecto.

4.1 Planificación

El proyecto se ha dividido en cinco grandes tareas:

- Análisis y Estado de la cuestión.

  Tarea en la que se analizan los objetivos del proyecto y el estado de la tecnología y los sistemas implicados en el mismo.

- Proceso de Extracción de la Información.

  Conjunto de tareas en la que se estudia el API de Twitter así como las tecnologías en las que se apoya.

  Dentro de esta tarea también se recogen las tareas del proceso de extracción de los *tweets* así como la decisión de qué colecciones de objetos *tweet* se adecúan mejora al objetivo del proyecto.

  Finalmente también pertenece a esta tarea general el desarrollo y utilización de la herramienta transformTwitterData encargada de realizar la transformación de los objetos *tweet* en formato json a Lenguaje Natural.

- Minería de sentimientos.

  Conjunto de tareas en las que se produce la creación de la ontología en la que se clasifican los *tweets:* creación y definición de términos, taxonomía, clústeres, patrones, etc.

- Interpretación y Evaluación de los Datos.

  Una vez traducido a Lenguaje Natural y creada la ontología, se procede con el análisis de las relaciones entre *tweets* y el proceso de extracción de información y evaluación de los resultados obtenidos.

- Presentación del Proyecto Fin de Carrera.

  La propia presentación del proyecto también implica un conjunto de tareas cuya planificación se recoge en esta tarea principal.

El diagrama Gantt muestra las cinco tareas presentadas así como su valoración en términos de tiempo y esfuerzo.

El proyecto es ejecutado por un Ingeniero Técnico como actor principal, la duración y el esfuerzo son equivalentes: duración de 127 días equivalentes a 1016 horas de una persona.

Además se incorpora la dedicación de del tutor del proyecto a modo de director del proyecto con una dedicación parcial del 10% del tiempo, equivalente a 12,7 días y 101,6 horas.

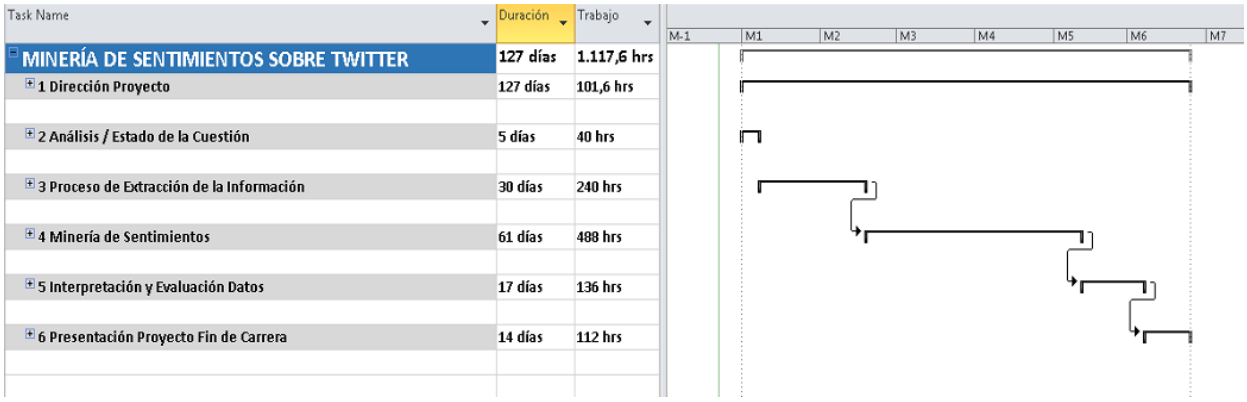

Ilustración 49 – Planificación Global

Entramos a continuación a detallar las tareas que componen cada tarea principal.

1. Dirección de proyecto.

   La ejecución del proyecto ha sido supervisada por la tutora del proyecto. Se incluye por lo tanto una tarea específica en el proyecto de coordinación, seguimiento y dirección del proyecto.

   La tarea es ejecutada a tiempo parcial por la tutora del proyecto.

Ilustración 50 – Dirección Proyecto

2. Análisis / Estado de la cuestión.

   La ejecución de esta tarea ha consistido en la recopilación de los objetivos del proyecto así como el estudio tanto de las tecnologías que se han de aplicar para su ejecución como de las tecnologías globales que aplican sobre: los procesos de minería de datos en general y la explotación de información de la red social Twitter.

El trabajo se ha apoyado en una gran cantidad de referencias fundamentalmente consultadas a través de internet.

3. Proceso de extracción de la información.

   En esta tarea se entra ya a fondo en el trabajo sobre Twitter para extraer la información de los *tweets* publicados en el perfil de la asignatura.

   En un primer momento se ha estudiado y analizado el API de Twitter así como las tecnologías en las que se sustenta.
   Durante el proceso de análisis también se han realizado aproximaciones a la mejor forma para obtener los *tweets* de la asignatura. Finalmente se ha decidido la utilización de la herramienta *API Console* proporcionada por el propio Twitter.

   Los trabajos de extracción realizados con *API Console* así como el estudio de la propia herramienta y la selección de las colecciones de *tweets* más idóneos dentro de los disponibles para explotar mediante el API, quedan reflejados en la planificación dentro de la tarea 3.1 Recolección de Datos.

   Una vez extraídos los *tweets* hay que resolver la problemática planteada para transformar su formato a un formato de Lenguaje Natural que posteriormente pueda ser tratado en el proceso de minería de sentimientos.

   Para el proceso de transformación finalmente se ha optado por el análisis y desarrollo de una utilidad a la que se ha dado el nombre de transofromTwitterData.

   Dentro de la tarea 3.2 Transformación se recogen las tareas que han guiado tanto el análisis, diseño y desarrollo de la herramienta como su utilización.

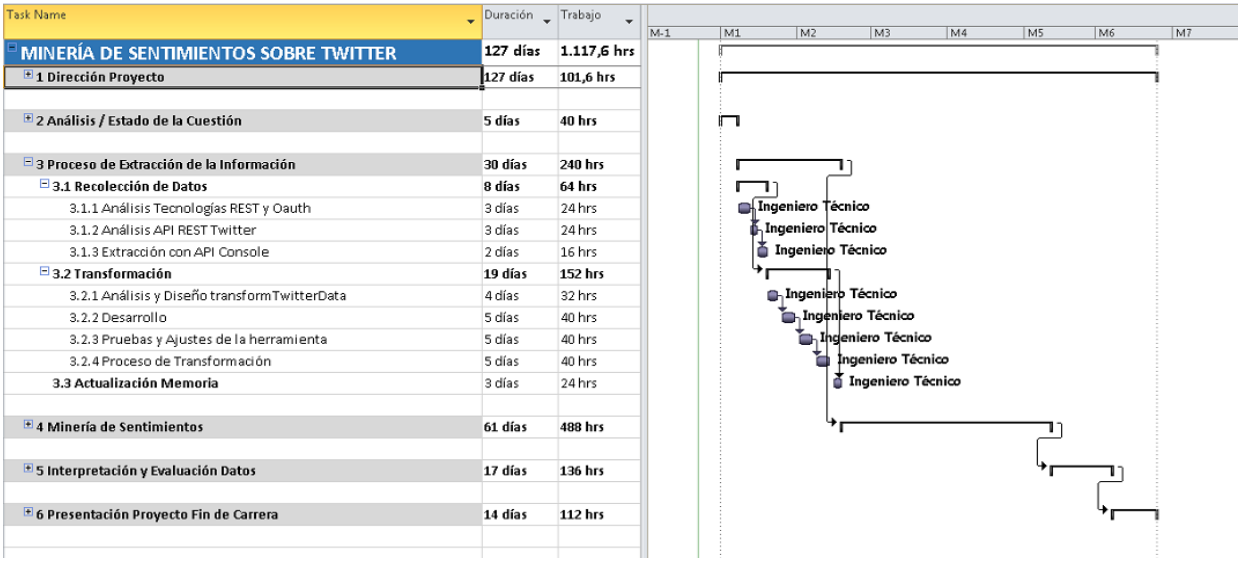

Ilustración 51 – Tareas proceso de extracción

4. Minería de sentimientos.

La minería de sentimientos es la tarea principal que se lleva más tiempo en el proyecto.

Con los datos ya transformados se ha trabajado, en primer lugar en el entendimiento de la herramienta knowledgeMANAGER y en segundo lugar en la tarea principal, creación de la ontología con todo lo que ello significa.

El enfoque a la hora de crear los patrones ha sido hacer una primera clasificación de los *tweets* en base a los criterios a medir. A partir de esa clasificación y con el análisis de los patrones que se detectan en cada tipo de expresión se ha procedido a la creación de los diferentes niveles de patrones.

El trabajo de creación y depuración de los patrones ha sido bastante intenso, de ahí que sean las tareas que se llevan más tiempo.
En el propio proceso de creación, se han realizado varias interacciones ya que según se iban probando los patrones se encontraban problemas y mejoras cuyo abordaje han ido enriqueciendo el modelo.

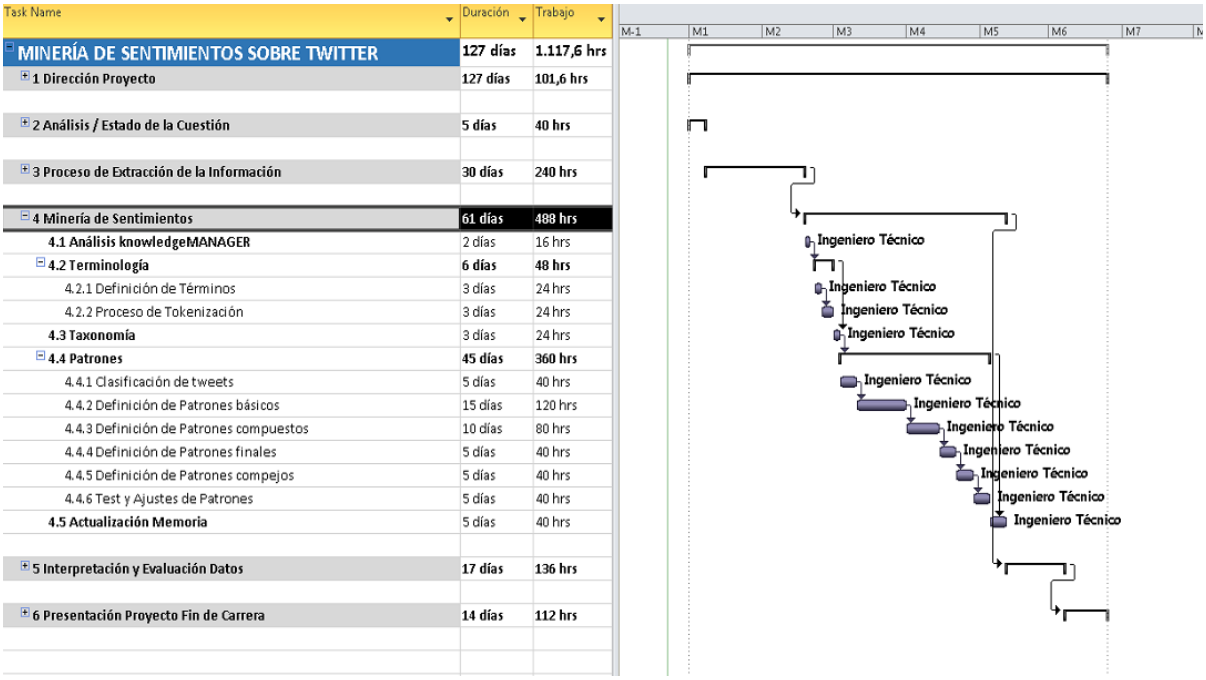

Ilustración 52 – Tareas minería de sentimientos

5. Evaluación e Interpretación.

Tarea fundamental dentro de la ejecución del proyecto.

En esta tarea se recogen todos los trabajos relativos a la interpretación de los resultados del proceso de minería de sentimientos.

El proceso de evaluación lo iniciamos mediante la creación de relaciones entre patrones y definición de meta-propiedades.

Tras el proceso de indexación se obtiene la instanciación de todas las meta-propiedades así como de los elementos que componen las relaciones. A partir de este resultado, se realiza un ejercicio de análisis de los resultados obtenidos mediante la elaboración de una serie de indicadores estadísticos.

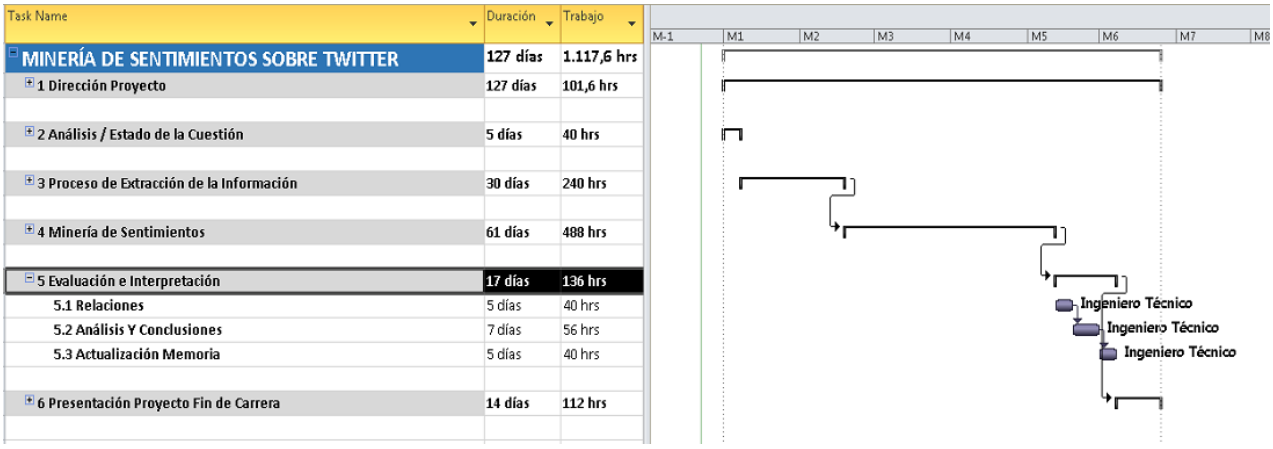

Ilustración 53 – Interpretación y Evaluación

6. Presentación del Proyecto Fin de Carrera.

   En todas las tareas se ha ido incorporando no sólo el trabajo técnico o de análisis e investigación sino que también se ha incluido el esfuerzo de ir incorporando el conocimiento y las conclusiones de cada paso a la memoria del proyecto.

   Aun así, queda una parte por finalizar y es la contemplada en esta tarea. No es más que la revisión general de la memoria así como las correcciones que surgen.
   Así mismo se incluyen dentro de esta tarea las tareas necesarias para desarrollar los apartados generales como los agradecimientos, resumen, etc. Y la adecuación del trabajo a las plantillas de documentación y presupuesto proporcionadas por la Universidad.

   Finalmente basado en el trabajo y en la propia memoria, está la tarea de elaboración de una presentación para ilustrar el trabajo en la defensa del proyecto.

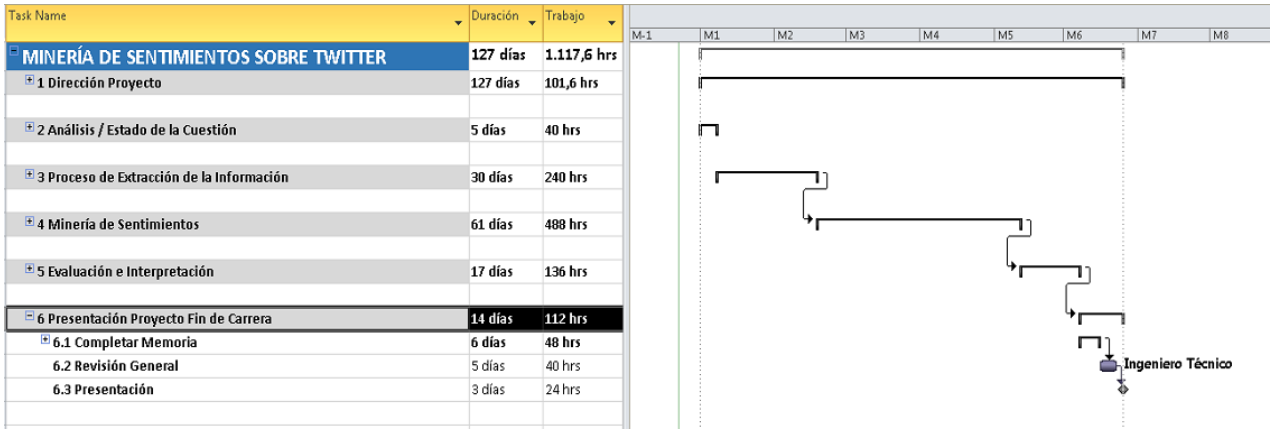

Ilustración 54 – Tareas Presentación y cierre del proyecto

4.2 Presupuesto

Los costes del proyecto se han dividido en los siguientes tipos.

- Costes de Personal.

  Costes de los perfiles necesarios para la realización del proyecto. Como se ha detallado en la planificación, se necesita la incorporación de un Ingeniero Técnico a dedicación completa y un Tutor / Director de Proyecto con dedicación del 10% durante toda la duración del proyecto.

  En la siguiente tabla se muestra el importe que alcanza este concepto: **24.176,88 €**

PERSONAL

Apellidos y nombre	Categoría	Dedicación (hombres mes) [a)]	Coste hombre mes	Coste (Euro)	Firma de conformidad
				0.00	
Anabel Fraga Vázquez	Directora Proyecto	0.77	4,288.54	3,319.74	
Antonio Martínez Rodríguez	Ingeniero Técnico	7.74	2,694.39	20,857.14	
				0.00	
		Hombres mes	**8.51**	**Total**	**24,176.88**

[a)] 1 Hombre mes = 131,25 horas. Máximo anual de dedicación de 12 hombres mes (1575 horas)
Máximo anual para PDI de la Universidad Carlos III de Madrid de 8,8 hombres mes (1.155 horas)

Tabla 33 – Costes Personal

- Costes de Hardware y Software.

  Costes de los elementos de hardware y software dedicados al proyecto.

  Durante la definición de la arquitectura se dieron alternativas de software libre en contraposición al software propietario utilizado en el desarrollo del proyecto.

  Debido a los bajos costes que representan el software dentro del volumen global, en la relación de software presupuestada se asume la utilización de software propietario.

  La siguiente tabla muestra el importe que alcanza este concepto: **384,67€**

  Para el coste imputable se ha considerado un plazo de depreciación de 60 meses y aplicado la siguiente fórmula.

  (A/B) x C x D
  Donde:

  A – Número de meses desde la fecha de la facturación del equipo.

Dado que el proyecto se planifica en más de 7 meses, para este cálculo se ha tomado como referencia 8 meses.

B – Periodo de depreciación.

Se ha tomado un periodo de 60 meses.

En el software ofimático Microsoft Office 365 Empresa no se ha considerado un periodo de depreciación ya que Microsoft da la opción de licenciar el producto por meses a razón de 10,7 € al mes (inferior si se contratan años enteros).

C – Coste sin IVA.

D - % de uso. En todos los casos del 100%.

HARDWARE Y SOFTWARE

Descripción	Coste (Euro)	% Uso dedicado proyecto	Dedicación (meses)	Periodo de depreciación	Coste imputable
Ordenador Portátil Profesional + Sistema Operativo Windows 8.1 Pro actualizable a Windows 10	900.00	100	8	60	120.00
Periféricos	35.00	100	8	60	4.67
Office 365 Empresa	56.00	100	8	1	56.00
Microsoft Project	110.00	100	8	60	14.67
knowledgeMANAGER	1000.00	100	8	60	133.33
Microsoft Visual Studio Profesional	420.00	100	8	60	56.00
				Total	**384.67**

Tabla 34 – Costes Hardware y Software

- Otros costes.

Como costes adicionales sólo se contempla la adquisición de material de oficina por valor de **50 €.**

Si a los costes anteriores le sumamos una estimación de costes indirectos del 20% del total, el presupuesto del proyecto asciende a 29.374 €.La siguiente tabla muestra el resumen de todos los costes presupuestados hasta el momento.

Presupuesto Costes Totales	Costes
Personal	27.177
Amortización	251
Subcontratación de tareas	0
Costes de funcionamiento	50
Costes Indirectos	4896
Total	29.374

Tabla 35 – Resumen de costes

Para finalizar el presupuesto, aplicamos dos conceptos más. Margen de beneficios que se desean obtener del proyecto y factor de riesgo que se asume.

Concepto	Factor corrector	Coste imputable
Margen Beneficios	20%	5.907
Factor de Riesgo	10%	2.937

Tabla 36 – Factores corrección presupuesto

Con lo que el presupuesto asciende a la cantidad de:

COSTE TOTAL	**€38.394**	sin IVA
	€46.457	con IVA

Tabla 37 – Coste Total

El presupuesto total de este proyecto asciende a la cantidad de CUARENTA Y SEIS MIL CUATROCIENTOS CINCUENTA Y SIETE EUROS.

Leganés a X de Octubre de 2015

El ingeniero proyectista

Fdo. Antonio Martínez Rodríguez

Capítulo 5

Extracción de la Información

A partir de este capítulo se detalla el proceso de implementación. Implementación iniciada con las tareas de extracción de la información.

El proceso de extracción de la información consiste en recoger los datos del perfil Twitter de la asignatura y su tratamiento (transformación y preparación) para convertirlos en información.

5. Extracción de la Información

El objetivo final del proyecto es la obtención de conocimiento a través de los datos aportados en el perfil Twitter de la asignatura.

Desde el punto de vista del aporte de valor, la relación es clara. Se parte de un volumen alto de datos, en nuestro caso datos obtenidos de una misma fuente, Twitter. Esos datos tratados de forma individual no aportan conocimiento al desarrollo de la asignatura.

Iniciamos la implementación para completar el primero de los objetivos secundarios marcados, la conversión de los datos en información.

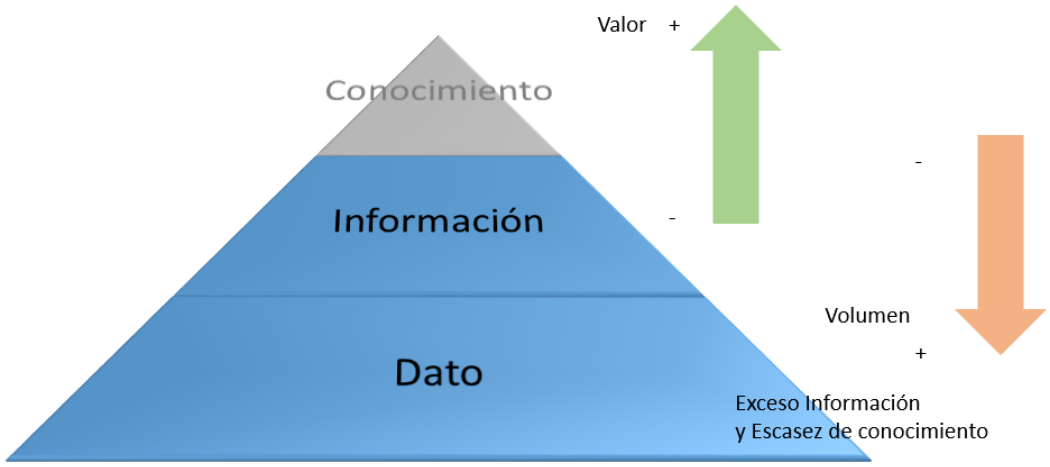

Ilustración 55 – Dato → Información

5.1 Recolección de Datos

Según lo establecido en la fase de análisis, el proceso de obtención de *tweets* de la asignatura se realizará mediante el uso de la herramienta API Console proporcionada por el propio Twitter a través de su página de desarrolladores [31].

En este apartado se detalla la implementación seguida para la obtención de los datos. En un primer lugar se establece el conjunto de datos que se han considerado relevantes para el estudio para a continuación explicar el uso de la utilidad API Console y su aplicación en el proyecto.

5.1.1 Recolección de Datos

El primer dilema que se plantea a la hora de recoger los datos de la cuenta es qué entendemos por todos los datos de la cuenta Twitter de la asignatura.

Parece claro que el objeto del estudio ha de ser analizar todos los *tweets* que se han intercambiado durante el curso en la cuenta de la asignatura, pero hay que matizar de qué *tweets* estamos hablando. Han de filtrarse aquellos *tweets* que se consideren relevantes para el estudio y obviar aquellos que no aporten información relacionada con el estudio.

- Menciones.

  Cuando un usuario escribe un *tweet*, salvo que se trate de un DM o mensaje directo, el *tweet* no tiene destinatario, pero es práctica común hacer mención al perfil sobre el que se está escribiendo el *tweet*.

  Es en este tipo de *tweets*, *tweets* con menciones al perfil *@miisi_uc3m* donde se encuentra el mayor volumen de información a incluir en el estudio.

Ilustración 56 –Menciones @miisi_uc3m

Procedemos con la obtención de las menciones mediante el uso de *API Console* a través del API REST de Twitter.

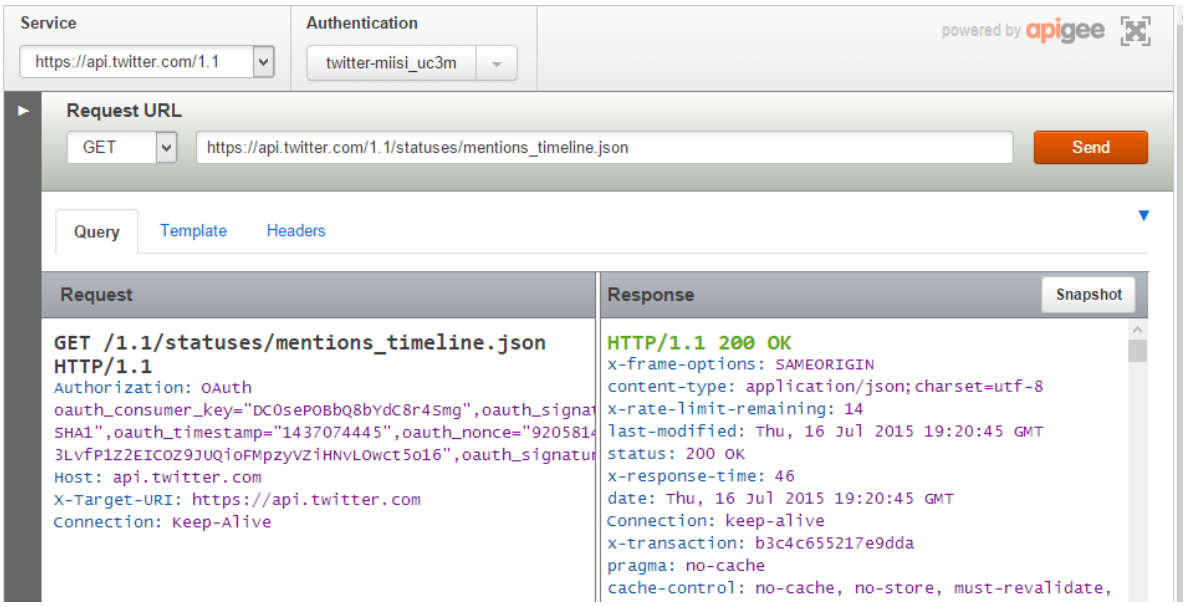

Ilustración 57 –API REST – Mentions misi_uc3m

Como puede apreciarse en la captura de pantalla anterior, siguiendo la filosofía de la tecnología REST, la solicitud es realizada mediante métodos HTTP estándar, en este caso GET. El GET se realiza sobre el objeto menciones.

Dado que nos encontramos autenticados con el usuario miisi_uc3m, existe autorización para obtener esta información. El método responde positivamente (*HTTP 200 OK)* e incluye en la respuesta todas las menciones realizadas a la cuenta.

La respuesta se realiza en formato *json* [20] y adicionalmente al texto del *tweet y al* usuario que lo ha enviado, incluye mucha más información: fecha y hora de envío, identificador del *tweet* en Twitter (generados con tecnología snowflake [35]), si es en respuesta a otro *tweet*, localización del *tweet*, localización del usuario, etc. En el apartado *5.2 Transformación* se aborda el proceso de tratamiento de toda esta información para hacerla *legible* de cara a su posterior procesamiento.


```
{
"created_at": "Sun May 31 18:21:23 +0000 2015",
"id": 605076621717053400,
"id_str": "605076621717053440",
"text": "@miisi_uc3m A mi me gusta más el tema de estrategia y transición de servicio, ya que son el punto inicial e
intermedio de todo el ciclo.",
"source": "<a href="http://twitter.com" rel="nofollow">Twitter Web Client</a>",
"truncated": false,
"in_reply_to_status_id": 573776088377262100,
"in_reply_to_status_id_str": "573776088377262080",
"in_reply_to_user_id": 3010797699,
"in_reply_to_user_id_str": "3010797699",
"in_reply_to_screen_name": "miisi_uc3m",
"user": {
"id": 2455299134,
"id_str": "2455299134",
"name": "roxana",
"screen_name": "roxana10720373",
"location": "",
"description": "",
```

```
"url": null,
"entities": {
"description": {
"urls": []
}
},
"protected": false,
"followers_count": 7,
"friends_count": 13,
"listed_count": 0,
"created_at": "Sun Apr 20 17:04:26 +0000 2014",
"favourites_count": 1,
"utc_offset": null,
"time_zone": null,
"geo_enabled": false,
"verified": false,
"statuses_count": 2,
"lang": "es",
"contributors_enabled": false,
"is_translator": false,
"is_translation_enabled": false,
"profile_background_color": "C0DEED",
"profile_background_image_url": "http://abs.twimg.com/images/themes/theme1/bg.png",
"profile_background_image_url_https": "https://abs.twimg.com/images/themes/theme1/bg.png",
"profile_background_tile": false,
"profile_image_url": "http://pbs.twimg.com/profile_images/605073705262292993/zIAob8j7_normal.jpg",
"profile_image_url_https": "https://pbs.twimg.com/profile_images/605073705262292993/zIAob8j7_normal.jpg",
"profile_banner_url": "https://pbs.twimg.com/profile_banners/2455299134/1433096023",
"profile_link_color": "0084B4",
"profile_sidebar_border_color": "C0DEED",
"profile_sidebar_fill_color": "DDEEF6",
"profile_text_color": "333333",
"profile_use_background_image": true,
"has_extended_profile": false,
"default_profile": true,
"default_profile_image": false,
"following": true,
"follow_request_sent": false,
"notifications": false
},
"geo": null,
"coordinates": null,
"place": null,
"contributors": null,
"is_quote_status": false,
"retweet_count": 0,
"favorite_count": 0,
"entities": {
"hashtags": [],
"symbols": [],
"user_mentions": [
{
"screen_name": "miisi_uc3m",
"name": "MIISI",
"id": 3010797699,
"id_str": "3010797699",
"indices": [
0,
11
]
}
],
"urls": []
},
"favorited": false,
```

"retweeted": false,
"lang": "es"

},

- Tweets enviados por el usuario.

  De los tweets relevantes para el estudio sólo quedan fuera de la categoría de menciones aquellos *tweets* enviados por el propio usuario de la cuenta en el que no se incluye una referencia explícita a la cuenta. Estos *tweets* también habrán de ser tenidos en cuenta.

  Por lo tanto, dentro del proceso de recolección de datos se extraerán todos los *tweets* del usuario.

  p.e.

MIISI @miisi_uc3m Mar 28

Alguno de vosotros piensa sacar el CISA?

Es un *tweet* enviado por @miisi_uc3m que non aparece dentro del listado de Menciones ya que el usuario no se menciona a sí mismo.

Como contraejemplo de un *tweet* enviado por el usuario y que aparece también en el listado de Menciones podemos ver el siguiente ejemplo donde el propio usuario incluye una mención a la cuenta de la asignatura.

Este *tweet* estará por lo tanto dentro del listado de *tweets* de menciones, por lo que se tendrá en cuenta no obtenerlo de este listado para evitar mensajes repetidos.

MIISI @miisi_uc3m May 27

@miisi_uc3m gracias a tod@s por la participación en la última clase, debates entretenidos y una presentación bien preparada!

Las siguientes clasificaciones de *tweets* contienen *tweets* visibles desde el cliente Twitter cuando se consulta la cuenta de la asignatura pero no se tendrán en cuenta para el estudio por las razones que se argumentan a continuación.

- Home del usuario.

  El usuario es *follower* de diferentes perfiles, perfiles de carácter generalista.
  Dado que no son *tweets* que aporten información sobre la asignatura y el desarrollo del mismo, estos *tweets* no han sido considerados.

  En la siguiente captura de pantalla pueden apreciarse algunos de los perfiles seguidos por el perfil de la asignatura y cuyos *tweets* han sido excluidos del análisis.

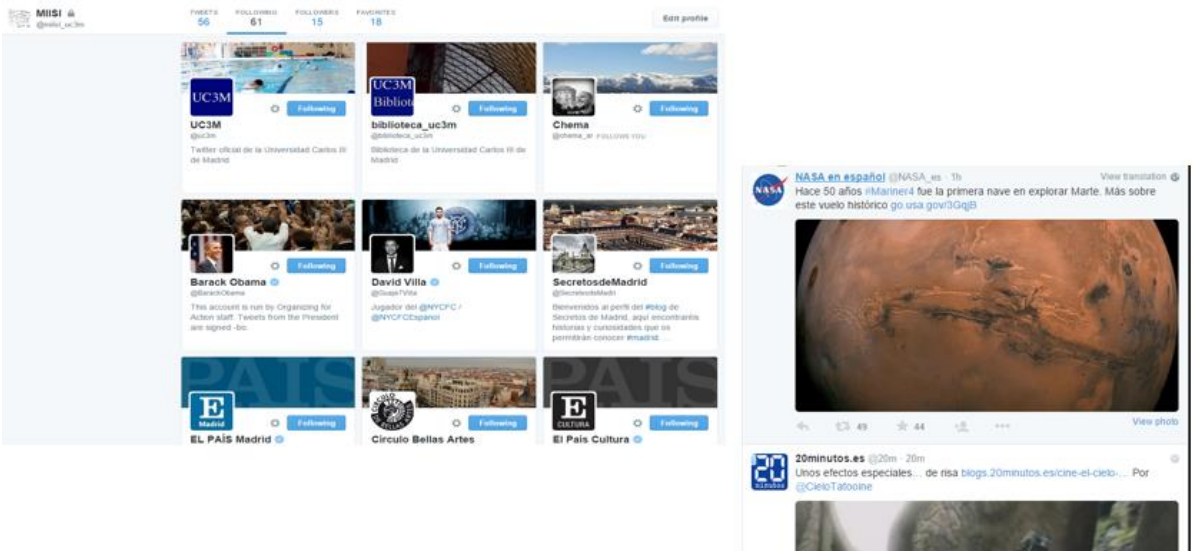

Ilustración 58 –Following miisi

- Notificaciones

  Las notificaciones que se van a recibir se configuran a nivel de perfil. La cuenta de la asignatura está configurada con los parámetros por defecto, básicamente se reciben notificaciones por cualquier tipo de interacción: *retweets* de mis *tweets*, *tweets* marcados como favoritos, etc.

  En la práctica del uso realizado por la cuenta, nos encontramos que los *tweets* englobados dentro de la categoría de notificaciones ya están previamente englobados en la categoría de menciones por lo que no se ha considerado su tratamiento al estar ya previamente incluido como mención.

- Mensajes directos

  No se ha realizado prácticamente uso de los DM o mensajes directos. Los únicos tres mensajes existentes carecen de relevancia.

Ilustración 59 –Mensajes Directos miisi_uc3m

Además de los *tweets* también se ha considerado necesaria la obtención de los *followers* de la cuenta en un listado separado ya que son datos que posteriormente se utilizarán en la minería de datos para tener trazabilidad y conocimiento de los usuarios que han participado en los diferentes hilos.

- Followers

Sólo se han considerado los *followers* aceptados por los gestores del perfil, sin tener en cuenta aquellas solicitudes de seguimiento que están pendientes de aceptación y que por otra parte no han participado en el intercambio de *tweets.*

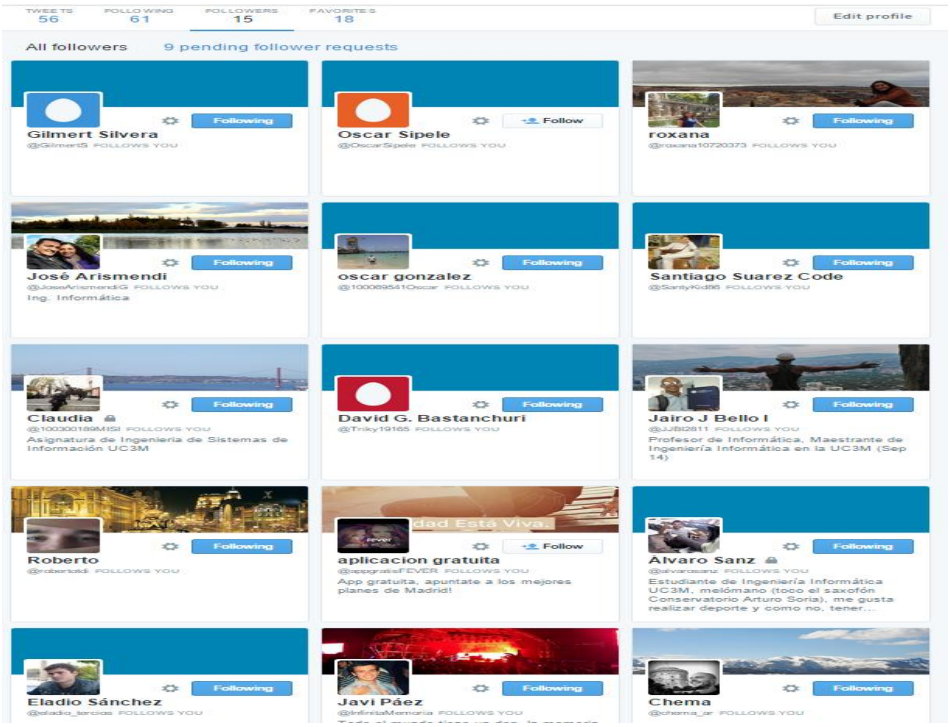

Ilustración 60 –Followers

5.1.2 API Console

A continuación se detalla el proceso de extracción de los datos.

La extracción se ha completado mediante la utilización de la aplicación *API Console*, accesible desde la propia página de desarrolladores de Twitter.

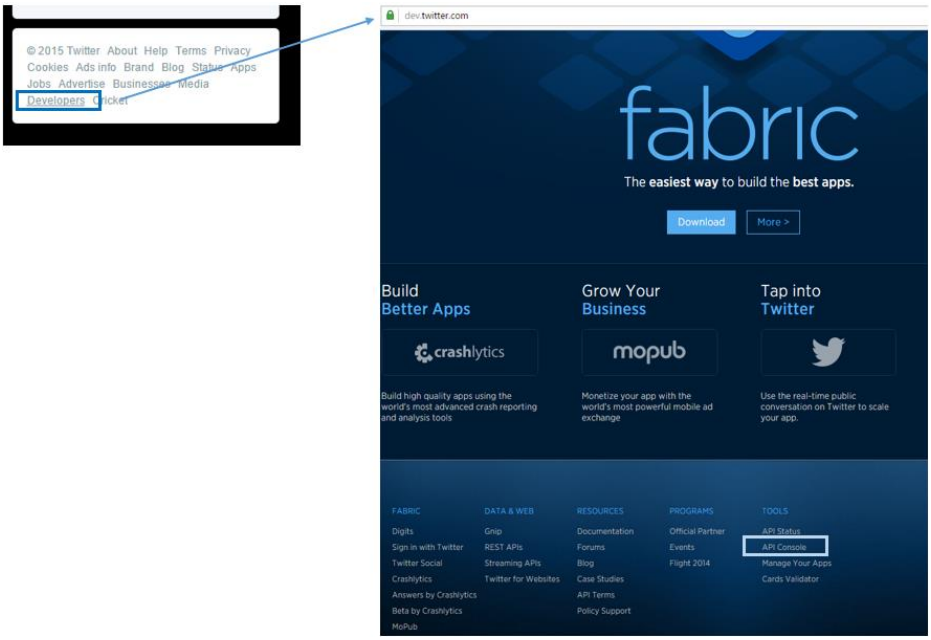

Ilustración 61 – Acceso a la aplicación API Console

Como no podía ser de otra forma, para la utilización del API hay que autenticarse de forma que Twitter autorizará las operaciones permitidas para el usuario. En el apartado 3.2.3.2 Tecnología OAuth ya se detalló la tecnología OAuth, veamos cómo se materializa su uso en el caso de esta aplicación.

- El Usuario se conecta a la aplicación mediante su perfil Twitter.

La aplicación informa al usuario de que la autenticación para su uso ha de realizarse a través de un perfil de Twitter. Será el propio Twitter quien de la autenticación y posteriormente, en base a esas credenciales, autorice el uso del API y adapte el resultado de retorno de las diferentes funciones.

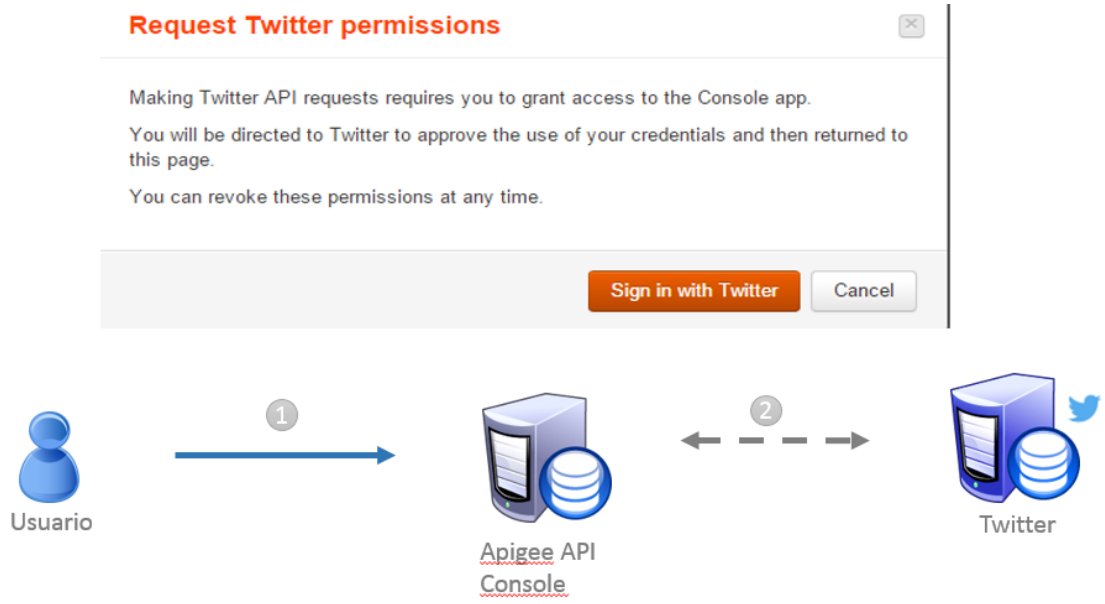

Ilustración 62 – Inicio Autenticación OAuth API Console – Twitter

- Twitter solicita las credenciales de usuario.

Como puede apreciarse en la *url* del formulario mostrado, es el propio Twitter quien solicita al usuario las credenciales para el acceso a la aplicación API Console informando de las autorizaciones que se están dando.

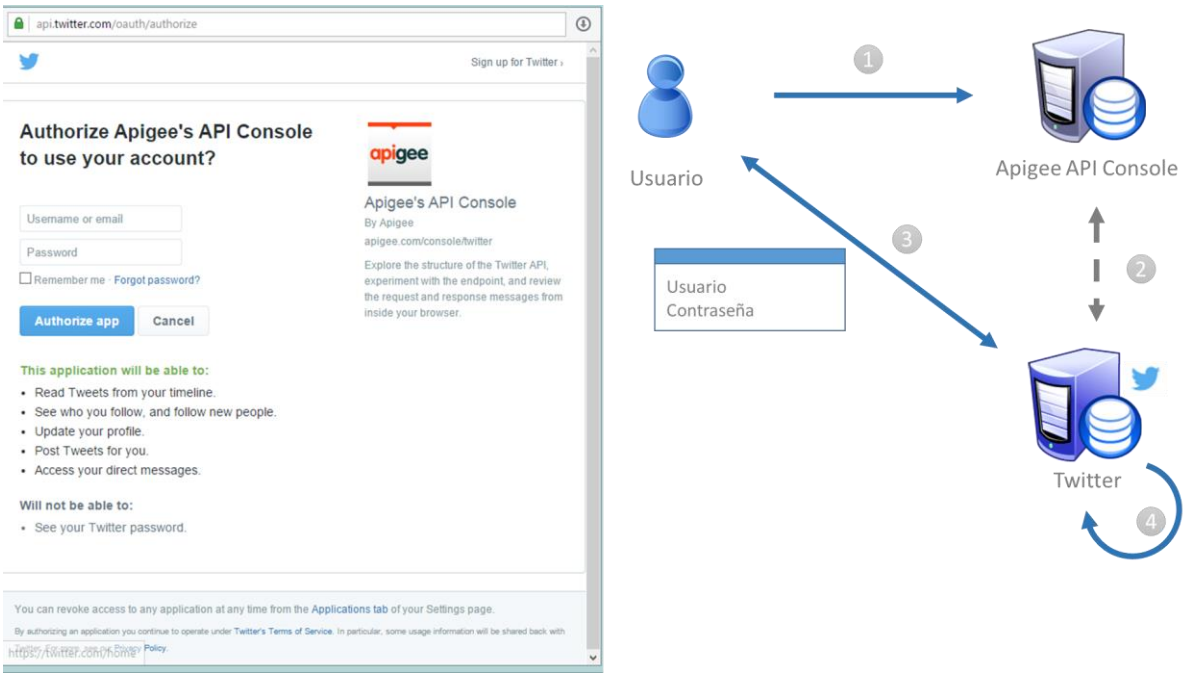

Ilustración 63 –Autenticación vía Twitter API Console

Tras autenticarse, Twitter direcciona al usuario a la aplicación, pero ya autenticado.

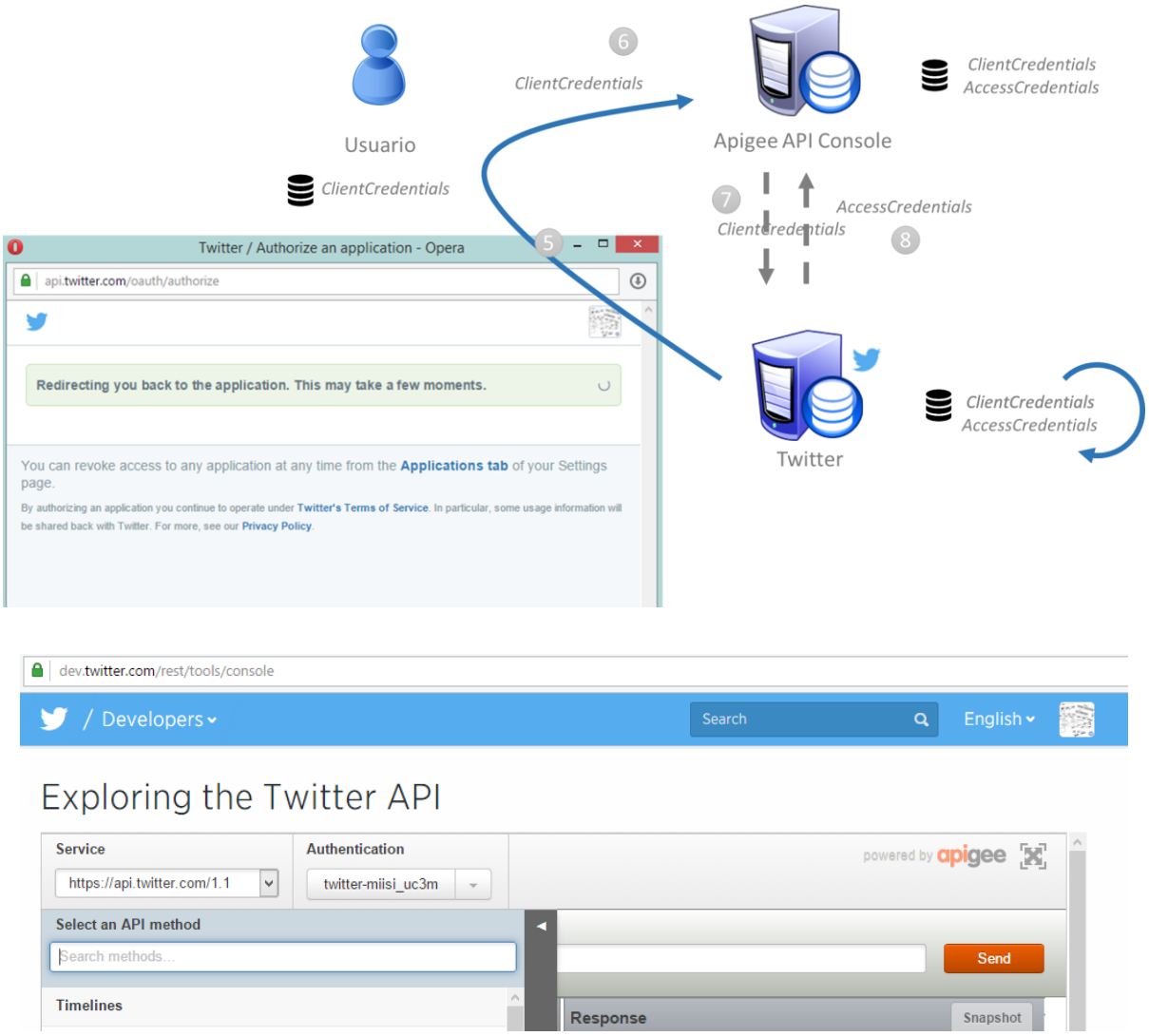

Ilustración 64 –Proceso de direccionamiento ya autenticado

Tal y como se ha detallado en el apartado anterior 5.1.1 Recolección de Datos la obtención de todos los *tweets* relevantes para el estudio se ha basado en dos conjuntos de *tweets*. Se han obtenido con el siguiente mecanismo.

- Menciones.

  Una vez autenticado, se hace la invocación REST *GET/statuses/mentions_timeline.json*.

```
GET /1.1/statuses/mentions_timeline.json HTTP/1.1
Authorization:
        OAuth
        oauth_consumer_key="DC0sePOBbQ8bYdC8r4Smg",oauth_signature_method="HMAC-
        SHA1",oauth_timestamp="1440523187",oauth_nonce="1364352045",oauth_version="1.0",
        oauth_token="3010797699-
```

3LvfP1Z2EICOZ9JUQioFMpzyVZiHNvLOwct5o16",oauth_signature="QMaHhSne%2FlbcewMW3a1y9R8nTpI%3D"
Host:
api.twitter.com
X-Target-URI:
https://api.twitter.com
Connection:
Keep-Alive

El resultado es almacenado en un fichero de texto en el formato *json* tal y como es devuelto por el API.

Ha sido suficiente con recuperar las doscientas últimas menciones ya que el número de *tweets* a recuperar es inferior a esa cifra.

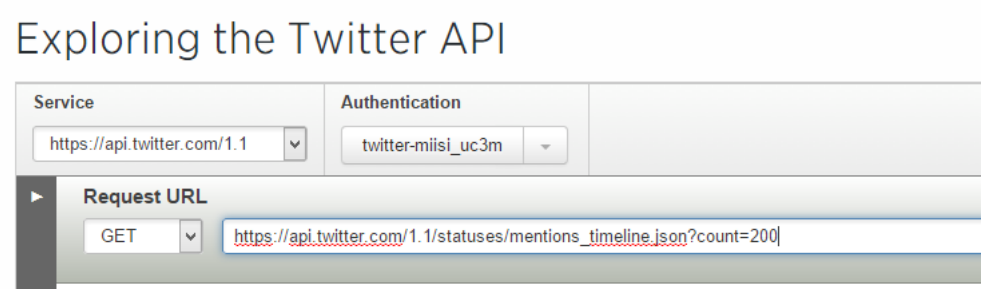

Ilustración 65 –Extracción de las Menciones

- *Tweets* del usuario.

Una vez autenticado, se hace la invocación REST GET/statuses/user_timeline.json.

GET /1.1/statuses/user_timeline.json?count=200 HTTP/1.1
Authorization:
OAuth
oauth_consumer_key="DC0sePOBbQ8bYdC8r4Smg",oauth_signature_method="HMAC-SHA1",oauth_timestamp="1440522849",oauth_nonce="1833336952",oauth_version="1.0",oauth_token="3010797699-3LvfP1Z2EICOZ9JUQioFMpzyVZiHNvLOwct5o16",oauth_signature="6G%2FGQm895c1oG%2FgCuE9VPPSIDuE%3D"
Host:
api.twitter.com
X-Target-URI:
https://api.twitter.com
Connection:
Keep-Alive

El resultado es nuevamente almacenado en un fichero de texto en el formato json.

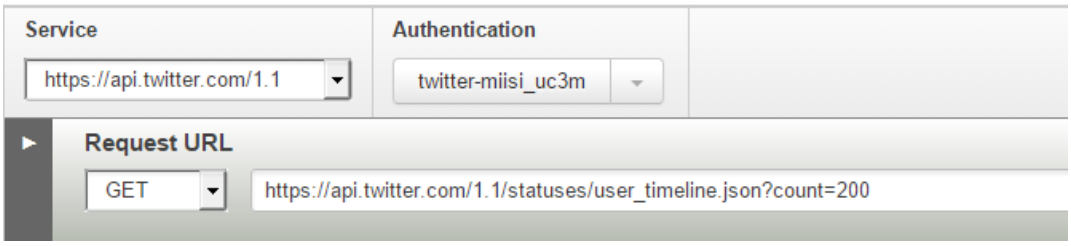

Ilustración 66 –Extracción tweets del usuario

- Followers.

  En este caso la solicitud REST realizada es GET/followers/ids.json.

```
GET /1.1/followers/ids.json HTTP/1.1
Authorization:
        OAuth
        oauth_consumer_key="DC0sePOBbQ8bYdC8r4Smg",oauth_signature_method="HMAC-
        SHA1",oauth_timestamp="1440524236",oauth_nonce="2766066576",oauth_version="1.0",
        oauth_token="3010797699-
        3LvfP1Z2EICOZ9JUQioFMpzyVZiHNvLOwct5o16",oauth_signature="%2BL8dLsmi7Enh
        UBRubAQ7qb0o1Sg%3D"
Host:
        api.twitter.com
X-Target-URI:
        https://api.twitter.com
Connection:
        Keep-Alive
```

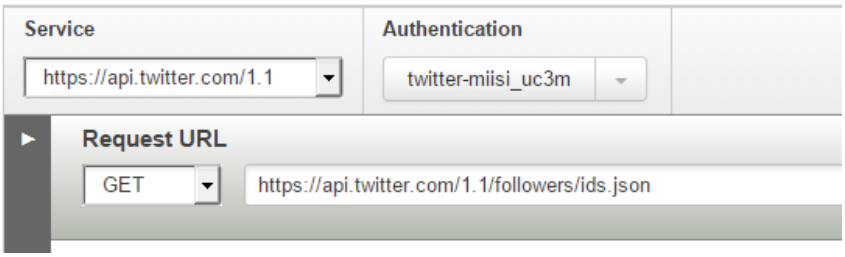

Ilustración 67 –Extracción followers del usuario

5.2 Transformación / Preparación

Esta etapa es la que convierte los datos en información.

En el punto anterior se ha descrito el proceso de descarga de *tweets* que en definitiva es el proceso de obtención de datos. Esos *tweets* por si mismos no representan información y más teniendo en cuenta que están en un formato no entendible por las personas. Es por ello que se hace imprescindible su transformación en lenguaje natural.

Repasemos mediante la siguiente ilustración cómo se enlazan los procesos de recolección y transformación.

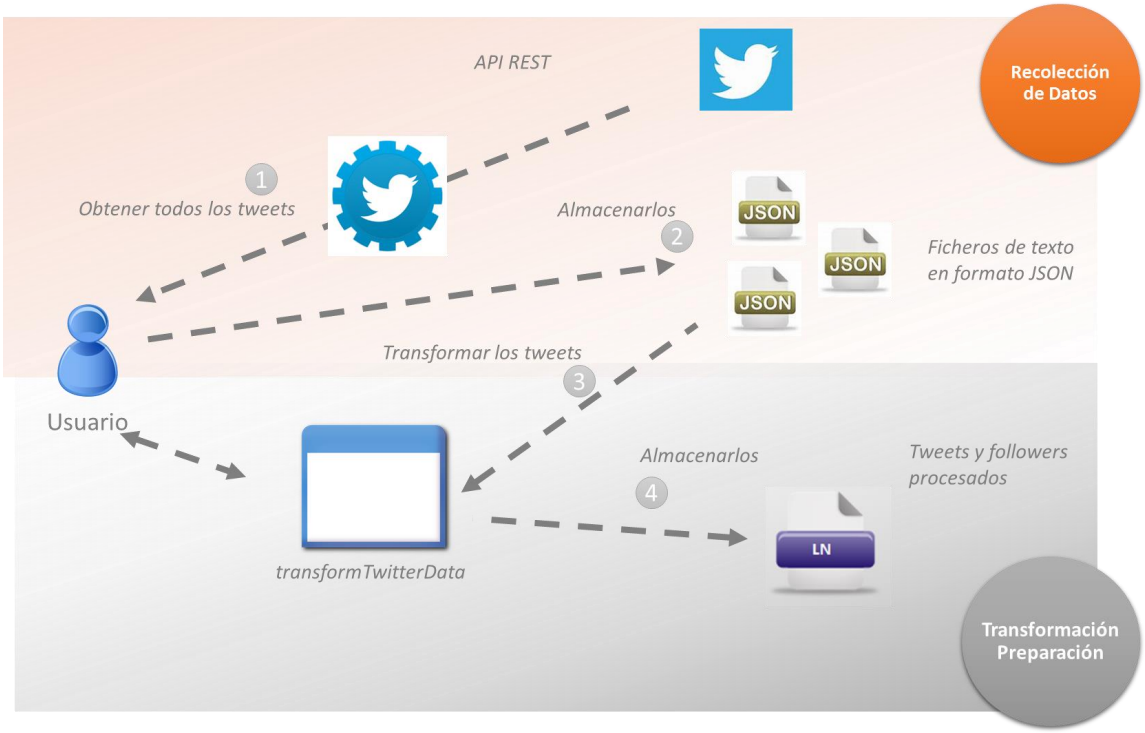

Ilustración 68 –Flujo Recolección y Transformación

1. Obtención de los *tweets* y *followers* mediante el API REST de Twitter.

2. Almacenamiento en ficheros con el formato original json.

Una vez que se dispone de los datos y mediante la utilización de la aplicación desarrollada *transformTwitterData:*

3. Procesamiento y transformación de datos a información

4. Almacenamiento del resultado del procesamiento.

La utilidad extraerá a partir del *tweet* en formato *json* la siguiente información.

- *Tweet ID*. Identificador único del *tweet*.
- Usuario. Identificador del usuario que ha escrito el *tweet*.
- Identificador del *tweet* al que responde. En el caso de que el *tweet* sea una respuesta a otro *tweet*, se incluye el *tweet ID* al que se responde.

- Texto del *tweet*. Contenido del *tweet*.

Y la convertirá a lenguaje natural.

> *"En el tweet: 605076621717053400, en respuesta al tweet: 573776088377262100, el usuario roxana10720373 comenta: miisiuc3m A mí me gusta más el tema de estrategia y transición de servicio, ya que son el punto inicial e intermedio de todo el ciclo."*

5.2.1 Funcionamiento y detalles de implementación

La aplicación se encarga de la transformación de los datos extraídos de Twitter mediante su API en un formato legible, entendible y procesable por *knowledgeMANAGER.*

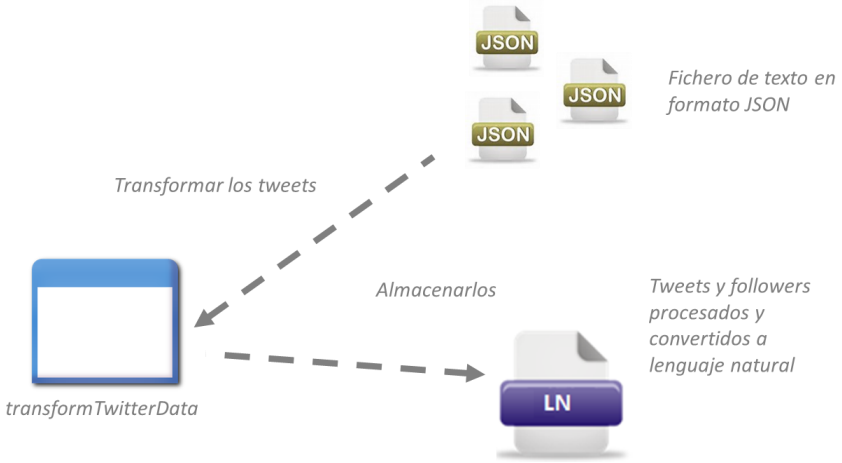

Ilustración 69 –Aplicación transformTwitterData

El interfaz gráfico de la aplicación se divide en dos áreas.

- Área de procesamiento y visualización

  En esta área de la aplicación es dónde se ejecuta el proceso de transformación, a su vez se divide en otras dos áreas.

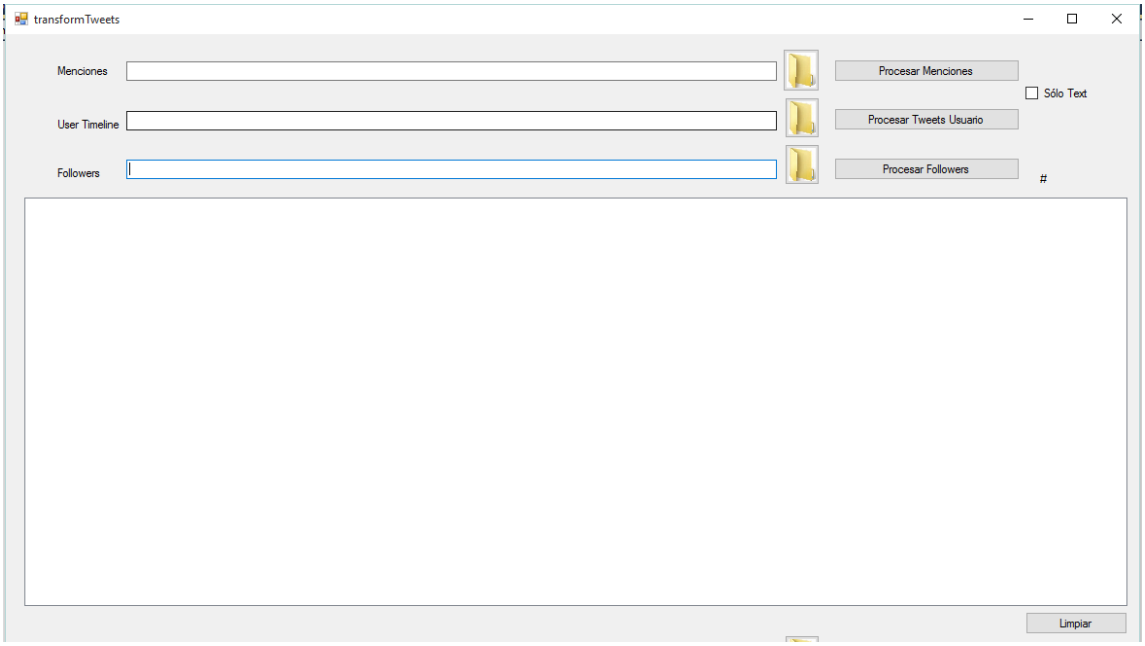

Ilustración 70 –Aplicación transformTwitterData – Área Procesamiento y visualización

- Selección del Fichero a Procesar.

  Se da la opción de navegar por el árbol de directorios del equipo para seleccionar el fichero extraído de Twitter.

  Debido a que cada conjunto de datos (menciones, *tweets* de usuario o *followers*) tiene un procesamiento diferente, el interfaz ofrece opciones específicas para cada uno de los elementos.

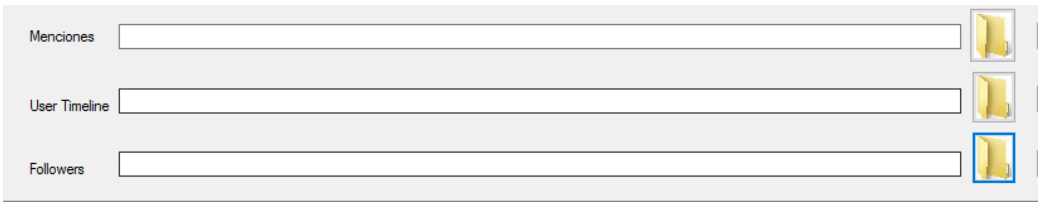

Ilustración 71 –Aplicación transformTwitterData – Área Procesamiento y visualización – Selección Ficheros

  Desde el punto de vista técnico la implementación de esta parte del interfaz no tiene complejidad, simplemente se apoya en .NET Framework para explotar las clases gráficas de selección de ficheros.

- Procesamiento.

  En esta parte es donde realmente se realiza toda la labor de Transformación y Preparación de los datos.

La transformación se realiza de forma diferente en función de los datos que se quieran procesar.

Podemos diferenciar en dos operaciones de procesamiento diferentes en base a los objetos que se trabajan.

1. Procesamiento de *tweets*.

   Engloba el procesamiento de *tweets* del listado de Menciones y de *tweets* del listado de *tweets* del usuario miisi_uc3m.

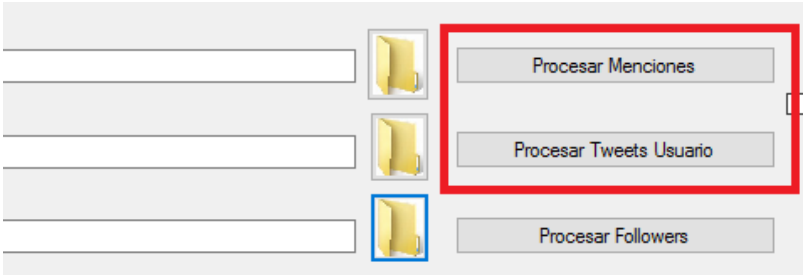

Ilustración 72 –Aplicación transformTwitterData – Área Procesamiento y visualización – Procesado de tweets

Ambos listados de *tweets* contienen objetos del tipo *tweet*. El objeto tiene un conjunto muy extenso de propiedades. En la documentación del API de Twitter pueden consultarse la composición completa del objeto [38]
Para el proceso de minería de datos se han considerado relevantes estas propiedades.

- Tweet ID. Identificador único del *tweet*.
- Usuario. Identificador del usuario que ha escrito el *tweet*.
- Identificador del *tweet* al que responde. En el caso de que el *tweet* sea una respuesta a otro *tweet*, se incluye el *tweet ID* al que se responde.
- Texto del *tweet*. Contenido del *tweet*.

En el objeto *tweet* estos datos se representados por sus correspondientes propiedades.

- Tweet ID: <id>
  Identificador único del tweet. Basado en tecnología *snowflake* [35].
- *Usuario: <user.name>*
  El usuario viene representado por su propio objeto de usuario (referencia completa del objeto usuario en la documentación de Twitter [44]). De cara al trabajo de este dato en el proyecto se ha considerado extraer el nombre/alias del usuario.

- Identificador del tweet al que responde: < in_reply_to_status_id_str>
  Si el tweet es respuesta de otro tweet nos encontraremos con esta propiedad rellena con el identificador único del tweet que se responde.

- Texto del tweet: <text>

La aplicación tiene una clase *tweet*, se van generando tantos objetos *tweet* como *tweets* hay en el fichero. Obviamente la clase únicamente tiene las propiedades anteriores y adicionalmente se ha añadido la fecha de generación del *tweet*.

transformTwitterData recorre el fichero (apoyándose en el namespace System.IO del .NET Framework) y lo analiza para ir extrayendo de cada *tweet* las propiedades.

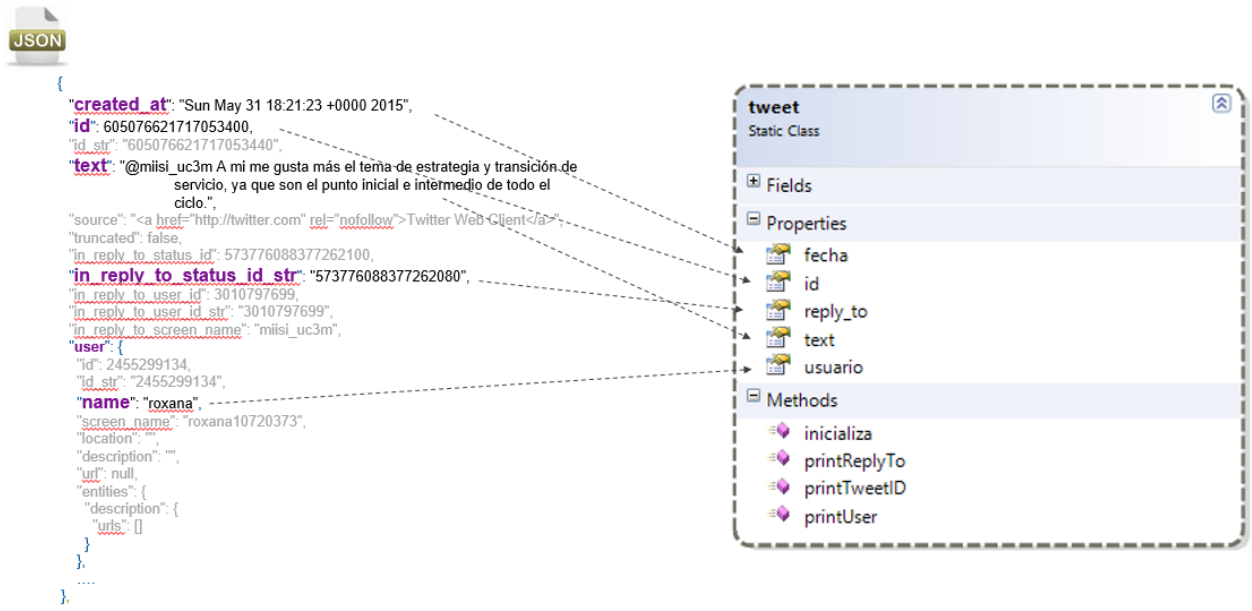

Ilustración 73 –Extracción del tweet a la clase tweet

Hasta este punto el procesamiento de Menciones y de *tweets* de usuario es el mismo ya que en ambos casos el fichero *json* contiene objetos *tweet* y en ambos casos las propiedades a considerar son las mismas.

Sin embargo, existe una diferencia a la hora de realizar la carga de los *tweets*. En el procesamiento de las Menciones se procesan y generan objetos *tweet* para cada uno de los *tweets* del fichero. En el procesamiento de los *tweets* del usuario no se cargan todos sino que se excluyen los siguientes.

- Retweets

  En el estudio preliminar de los datos se aprecia que el usuario misi_uc3m acostumbra a hacer *retweets* de *tweets* de otros perfiles que si bien están relacionados con la

asignatura, no son más que referencias y elementos ajenos los intercambios de información y opiniones que se producen en los *tweets*.

- *Tweets* del usuario miisi_uc3m en el que se menciona a sí mismo.

  Se excluyen ya que dichos *tweets* ya se encuentran en el listado de *tweets* de Menciones.

En la etapa de Recolección de Datos no fue posible realizar esta diferenciación ya que la recolección se hace en bruto de todos los *tweets*. Siendo responsabilidad de esta etapa el filtrado de los datos relevantes.

2. Procesamiento de *followers*.

   El procesamiento del fichero *json* de *followers* exige de un tratamiento previo a su gestión por transformTwitterData.

   El fichero obtenido en la fase de recolección contiene los identificadores únicos de usuario de todos los *followers*, pero de cara a hacer más legible y fácilmente procesable la información, se ha considerado necesario referenciar a dichos usuarios por su nombre/alias. De hecho la extracción de datos del *tweet* no está teniendo en consideración el identificador del usuario sino su nombre o alias.

   La extracción del fichero json final de *followers* implica nuevamente la utilización del API REST de Twitter. En este caso se utiliza la operación GET/friendships/lookup.json pasando como parámetro todos los identificadores de usuario.

```
GET
/1.1/friendships/lookup.json?screen_name=miisi_uc3m&
user_id=349774565%2C3088932533%2C2455299134%2C429073
499%2C3015884462%2C3027063785…. HTTP/1.1
Authorization:
        OAuth
        oauth_consumer_key="DC0sePOBbQ8bYdC8r4Smg",oauth_signature_method="HMA
        C-
        SHA1",oauth_timestamp="1440538553",oauth_nonce="3206302906",oauth_version="
        1.0",oauth_token="3010797699-
        3LvfP1Z2EICOZ9JUQioFMpzyVZiHNvLOwct5o16",oauth_signature="iQvOgawku9O
        v4sv%2FA5DghtTRrco%3D"
Host:
        api.twitter.com
X-Target-URI:
        https://api.twitter.com
Connection:
        Keep-Alive
```

El *json* obtenido con esta invocación retorna tantos objetos de usuario como *followers* tenga la cuenta.

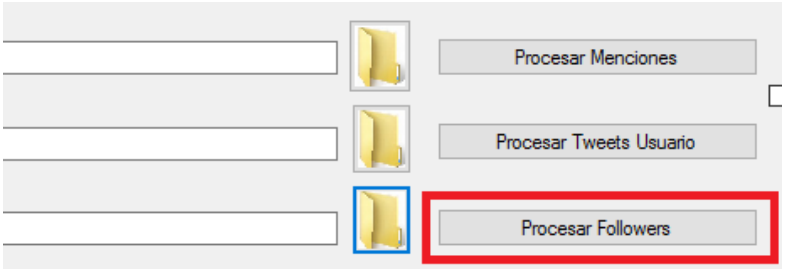

Ilustración 74 –Aplicación transformTwitterData – Área Procesamiento y visualización – Procesado de followers

El procesamiento a partir de aquí es el mismo que el detallado para los objetos *tweet* con la diferencia de que en este caso la única propiedad que se tiene en consideración es el nombre o alias del usuario, ubicado en la propiedad <screen_name>.


```
{
  "name": "Oscar Sipele",
  "screen_name": "OscarSipele",
  "id": 3088932533,
  "id_str": "3088932533",
  "connections": [
    "followed_by"
  ]

},
```


- Visualización

Para finalizar con el procesamiento, los *tweets* o *followers* procesados se muestran por pantalla ya transformados, previamente a ser almacenados.

- *Tweet*

Se muestra una línea con el formato convenido por cada *tweet*.

> *"En el tweet: <tweetID>, en respuesta al tweet: <tweetID>, el usuario <usuario> comenta: <texto del tweet>"*

Si el *tweet* no es una respuesta a otro *tweet*, no se añade el texto "en respuesta al *tweet <tweetID>".*

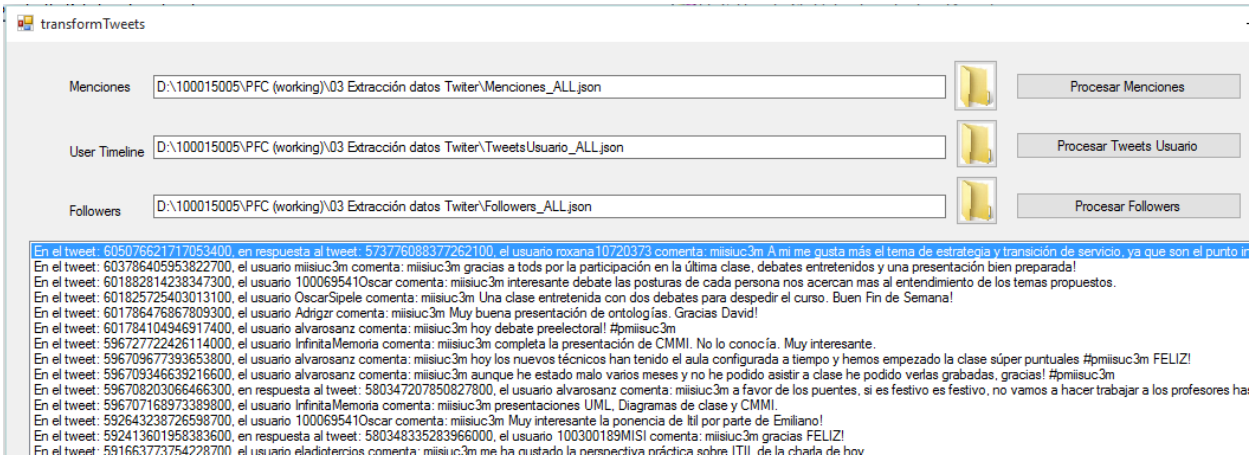

Ilustración 75 –Visualización del procesamiento de tweets

- Seguidores.

  Se muestra una línea con cada *follower*. En este caso sólo se muestra el nombre del usuario.

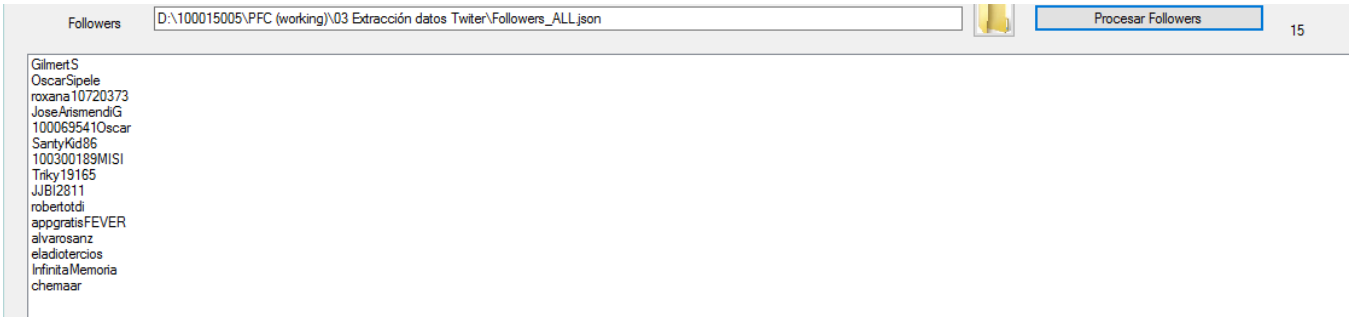

Ilustración 76 –Visualización del procesamiento de followers

- Generación de los datos de salida.

  Una vez que se han procesado los datos, se puede proceder con la exportación de los mismos a un fichero de texto con el formato legible definido.
  El fichero de texto será utilizado en la posterior fase de minería.

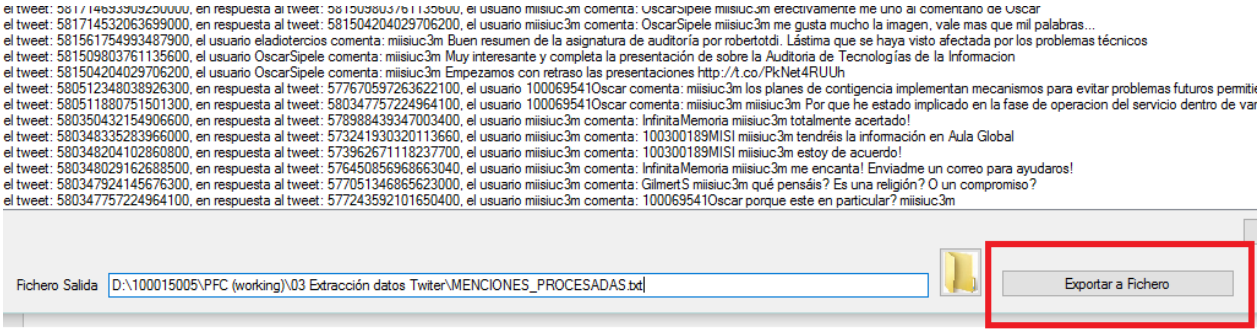

Ilustración 77 –Exportar los resultados a fichero

Capítulo 6

Minería de Sentimientos

Tomando como salida la Información del proceso de extracción de información, en este apartado se detalla el proceso de minería de sentimientos basado apoyado por el desarrollo de una ontología y su procesado con *knowledgeMANAGER.*

6. Minería de Sentimientos

Partiendo de la Información obtenida en la fase de extracción y transformación/procesamiento, queda la etapa final consistente en procesar dicha información y convertirla en conocimiento.

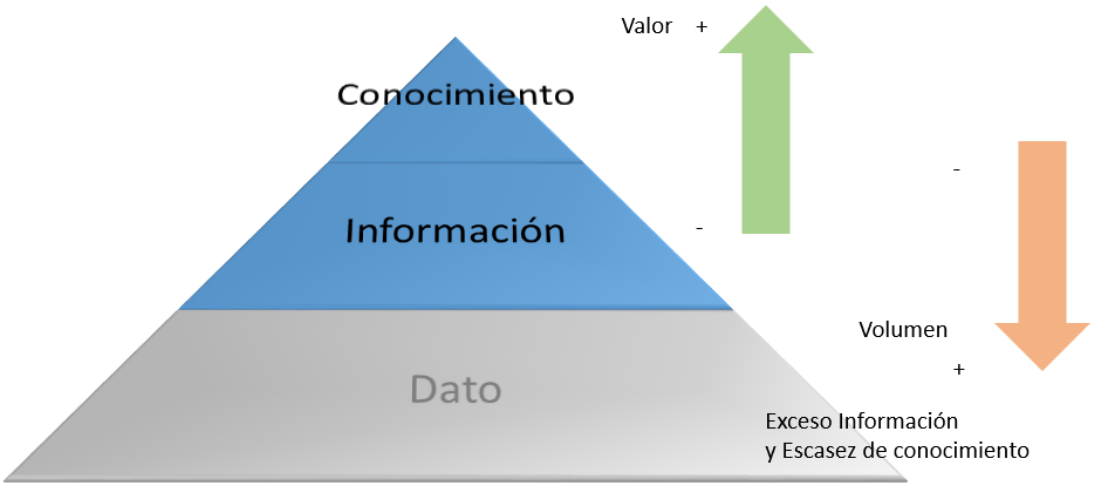

Ilustración 78 – Información → Conocimiento

Como se detalló en el análisis, para el desarrollo del presente proyecto se ha optado por implementar un desarrollo del proceso de minería de datos asistido por ontologías. Esto es, a partir de la información disponible se desarrolla una ontología en la que se conceptualiza el universo de *tweets* de la asignatura objeto del estudio y se utiliza la ontología para analizar propiedades, relaciones y extraer los resultados.

6.1 knowledgeMANAGER

Para el desarrollo de la ontología se ha utilizado la herramienta knowledgeMANAGER de la compañía The REUSE Company [40]. La herramienta está diseñada y desarrollada para el trabajo con ontologías. Tiene por lo tanto herramientas para la creación de los diferentes elementos: términos, tokens, patrones, clústeres, etc., así como herramientas para la definición y trabajo con patrones.

En el presente trabajo se detallarán los pasos seguidos dentro de la herramienta para la consecución de los objetivos del proyecto. En la siguiente referencia puede encontrarse el manual de usuario completo de la misma para ahondar en los detalles y capacidades extra [45]

La elaboración de la ontología se realizará en cuatro fases.

- Fase de definición de la terminología

  En esta fase se definen los términos que componen el dominio del estudio.

  Esta fase se completará con la opción de manejo de terminología de knowledgeMANAGER.

Ilustración 79 –knowledgeMANAGER gestión de términos

- Fase de Taxonomía

  Se definen agrupaciones de términos, y reglas de *tokenización* (análisis y conversiones léxicas de términos).

  Para la consecución de esta fase se accede a dos opciones de la herramienta.

  - Tokenización

    En la misma opción de terminología se definen y ajustan las reglas de token.

Ilustración 80 –knowledgeMANAGER Taxonomía - Tokenización

  - Clústeres

    Esta opción permite definir la agrupación de términos en base a la semántica que los relaciona.

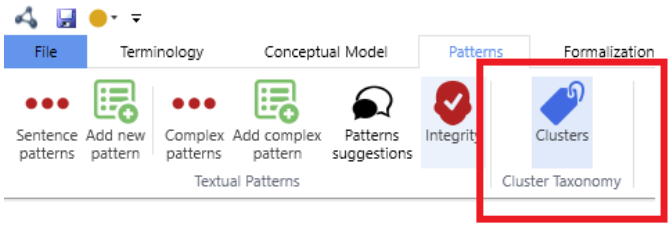

Ilustración 81 –knowledgeMANAGER Taxonomía - Clústeres

- Fase de definición de creación de patrones

  Se crea una jerarquía de patrones y sub-patrones hasta llegar a las expresiones finales.
  Los patrones definen la sintaxis de los *tweets*.

  La herramienta proporciona un conjunto de acciones de creación, revisión, agrupación de patrones.

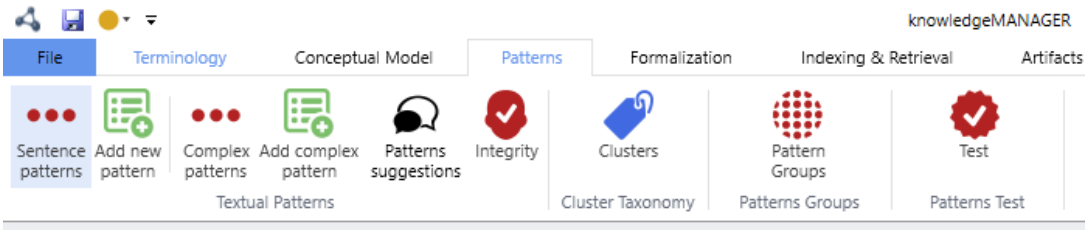

Ilustración 82 –knowledgeMANAGER Patrones

- Fase de formalización

  Se dota de semántica a los *tweets* mediante el establecimiento de relaciones entre patrones y la definición de meta-propiedades que representan semánticamente al *tweet* (positivo, negativo, etc.).

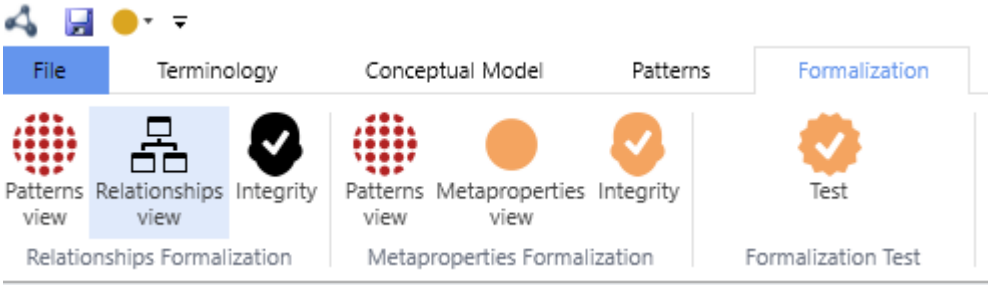

Ilustración 83 –knowledgeMANAGER Formalización

La siguiente ilustración muestra gráficamente cómo se compone la ontología. Del elemento más básico, los términos, a la fase más avanzada, la formalización.

Estas fases de la ontología se basan en las que propone la propia herramienta [45] con alguna adaptación al caso que nos ocupa.

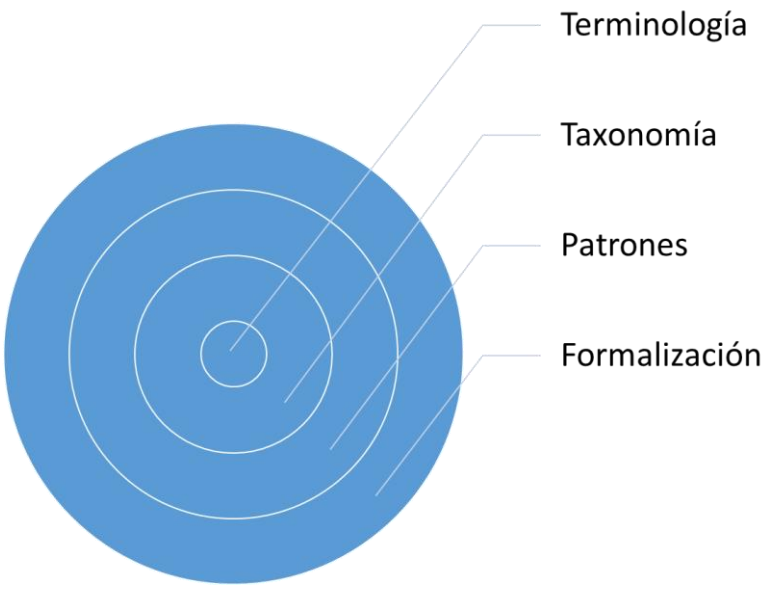

Ilustración 84 –Ontología del proyecto

6.2. Terminología

El primer paso en la definición de la ontología consiste en definir los términos individuales que componen el vocabulario sobre el que se irán construyendo el resto de los objetos.

La herramienta trae precargada de partida una base de datos con una gramática de lenguaje natural en castellano por lo que todos los términos y la taxonomía de los mismos viene precargada. Es decir, el reconocimiento de los textos en castellano que tienen los *tweets* no es necesario definirlos (determinantes, artículos, nombres, verbos, etc.).

En este punto por lo tanto lo que se ha realizado es incluir los términos específicos de Twitter.

- Identificadores de *tweets*.

  "En el tweet: 569445106283106300, el usuario miisiuc3m comenta: Hola! comentad vuestras opiniones de las presentaciones. un abrazo!"

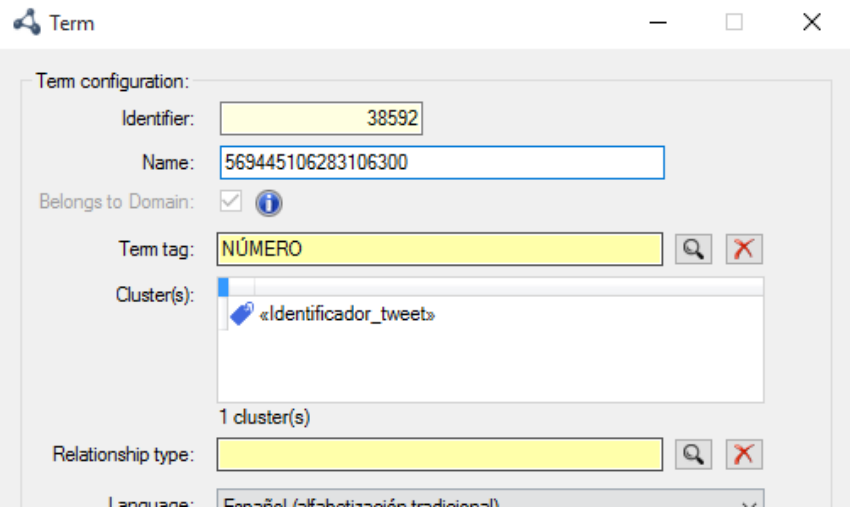

Ilustración 85 –Ejemplo de término definido en la ontología del proyecto

- Estándares y metodologías.

  Forma parte del estudio y se mencionan con cierta frecuencia nombres de estándares o metodologías como ISO, ITIL, COBID, CMMI.

  "En el tweet: 596727722426114000, el usuario InfinitaMemoria comenta: miisiuc3m completa la presentación de CMMI. No lo conocía. Muy interesante."

  Algunos términos no estaban definidos por lo que se ha procedido con su incorporación.

- *Hashtags* utilizados.

  Al inicio del curso se facilitó a los usuarios una serie de *hashtags* que podían utilizar cuando publicaran un *tweet*. De esta forma se pretendía facilitar el análisis de sentimientos del *tweet*.

hashtag	**Carácter del *tweet***
#pmiisuc3m	*Tweet* positive
#nmiisuc3m	*Tweet* negativo
#imiisuc3m	*Tweet* solicitando información
#qmiisuc3m	*Tweet* para preguntas
#cmiisuc3m	*Tweet* para realizar quejas
#smiisuc3m	*Tweet* sarcástico

Tabla 38 – hashtag de la asignatura

Si bien no han sido muy utilizados, sí que existen algunos *tweets* que incluyen alguno de los *hashtags* por lo que han incluido como términos.

"En el tweet: 577863532840124400, el usuario alvarosanz comenta: miisiuc3m gracias al material subido y los compañeros no he perdido el hilo de las clases después de perder dos por trabajo #pmiisuc3m"

- Usuarios

  Nombre de los usuarios que han participado, bien siendo seguidores de la cuenta, bien enviando *tweets*.

6.3. Taxonomía

> **Taxonomía.** [46]
>
> (Del gr. τάξις, ordenación, y -nomía).
>
> **1.** f. Ciencia que trata de los principios, métodos y fines de la clasificación. Se aplica en particular, dentro de la biología, para la ordenación jerarquizada y sistemática, con sus nombres, de los grupos de animales y de vegetales.
>
> **2.** f. clasificación (‖ acción y efecto de clasificar).

La fase de Taxonomía consistirá en clasificar y agrupar los términos (*tokens*) así como en aplicarles algunas reglas de análisis léxico (*tokenización)*.

A la hora de plantear la clasificación y ordenación de los términos que se utilizan en los *tweets* surgieron dos posibilidades.

- Definición de lo que la herramienta llama tags que son etiquetas sobre las que agrupar términos.

  La herramienta ya venía precargada con un conjunto de *tags* relativos a la lengua castellana. La incorporación de etiquetas nuevas a la estructura ya implementada supuso algún problema de rendimiento a la hora de analizar patrones por lo que se decidió realizar la agrupación en base a otro elemento de agrupación, en este caso específico para taxonomía.

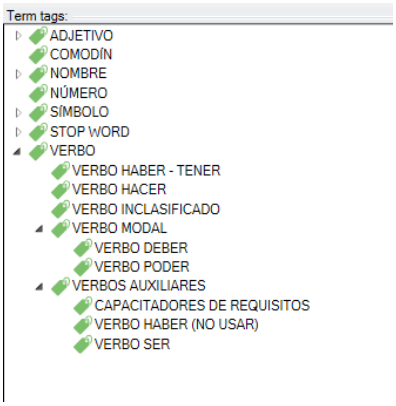

Ilustración 86 – tags definidos de base en knowledgeMANAGER

- Clústeres.

  La herramienta ya incluye un conjunto de clústeres (o contenedores) definidos, la incorporación de nuevos clústeres no supone ningún problema sobre el rendimiento de la herramienta en los análisis posteriores.
  Para diferenciarlos de los ya implementados se definió una nueva estructura de contenedores.

  Hay que tener en cuenta que los clústeres se definen de forma jerárquica, en forma de árbol. Se ha creado el elemento raíz *tweet* y a partir de ahí cuelgan los diferentes contenedores.

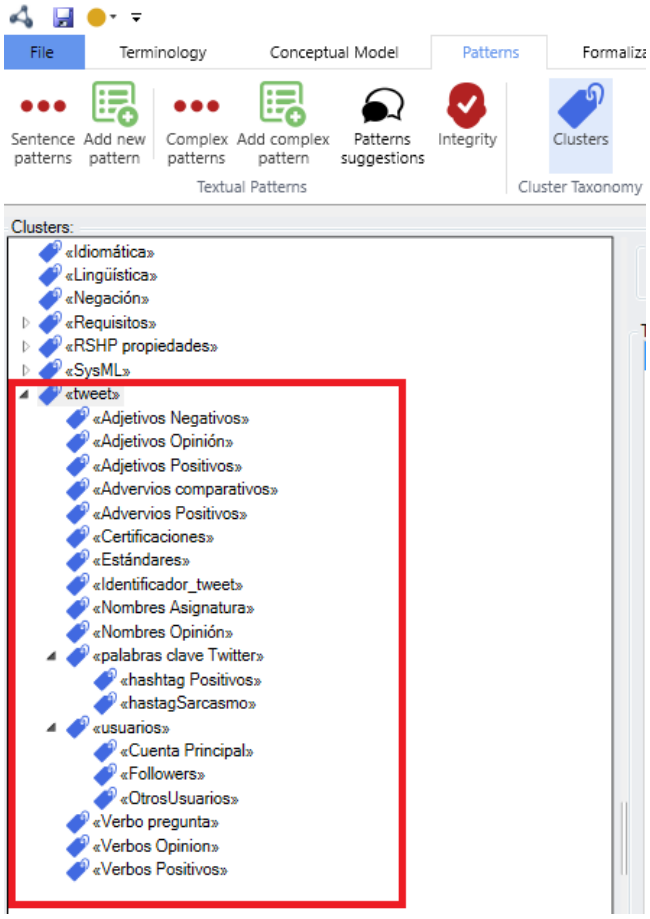

Ilustración 87 – cluster definidos para el proyecto

Una vez creados los contenedores, se ha procedido a asignar los términos utilizados en los *tweets* dentro del *cluster* que corresponde.

- Cluster: Adjetivos Positivos.

  Acertado, buen, buenísima, bueno, entretenido, interesante, preparado, etc.

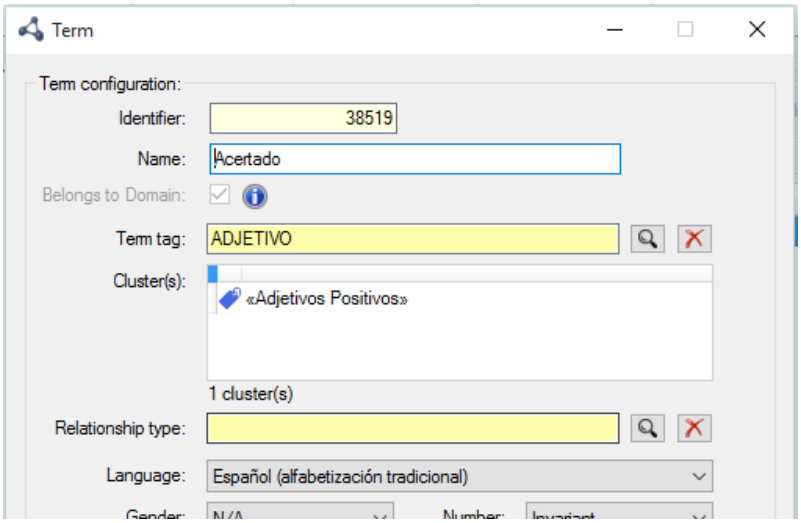

Ilustración 88 – Término asignado a un determinado cluster

- Cluster: usuarios\followers.

  Alvarosanz, Chemaar, Eladiotercios, Gilments, etc.

- Etc.

El objetivo de estos *clústeres* es formar parte posteriormente de los ítems que definen los patrones de forma que cuando se define un patrón, en lugar de hacer referencia a un término específico, se hace referencia al *cluster* que contiene términos de ese tipo.

Finalmente para finalizar con las tareas de la fase de Taxonomía se han definido algunas reglas de *tokenización*. Antes de nada, expliquemos en qué consisten dichas reglas a nivel general.

Dentro del reconocimiento del todo lenguaje natural existen acrónimos, abreviaturas o léxicos que se utilizan comúnmente y que se corresponden con términos concretos.

knowledgeMANAGER está preparado para procesar esos términos y transformarlos en el término equivalente. La siguiente ilustración muestra una regla creada para convertir acrónimos. En este caso se traduce “Mª” a su equivalente María.

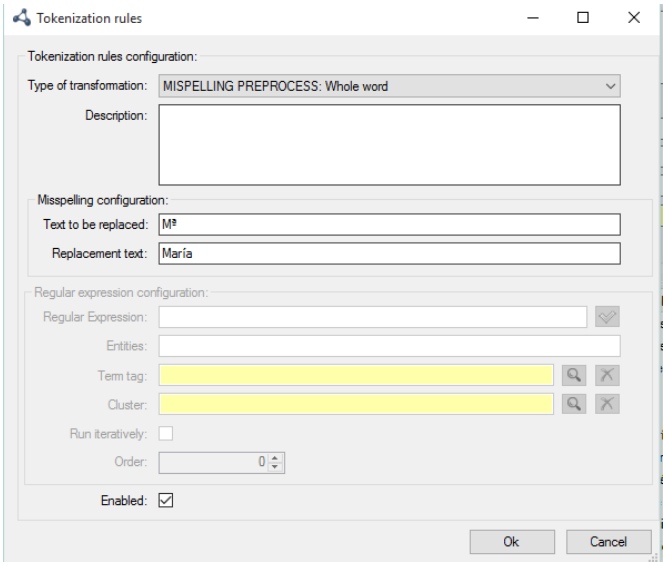

Ilustración 89 – Reglas de tokenización – Conversión de acrónimos

Debido al léxico particular de Twitter, durante la fase de definición y creación de patrones se encontró algún problema con usuarios cuyo nombre se compone de un conjunto de números en primer lugar y luego letras. Sobre estos nombres se ha hecho necesaria la definición de reglas de transformación de términos.

Veámoslo con un ejemplo.

1. Nos encontramos con un usuario cuyo nombre de perfil es 100015005Antonio.

2. Dentro del reconocimiento de lenguaje natural la gramática tiene definida una regla de forma que reconoce como un número cualquier relación de n caracteres numéricos consecutivos.

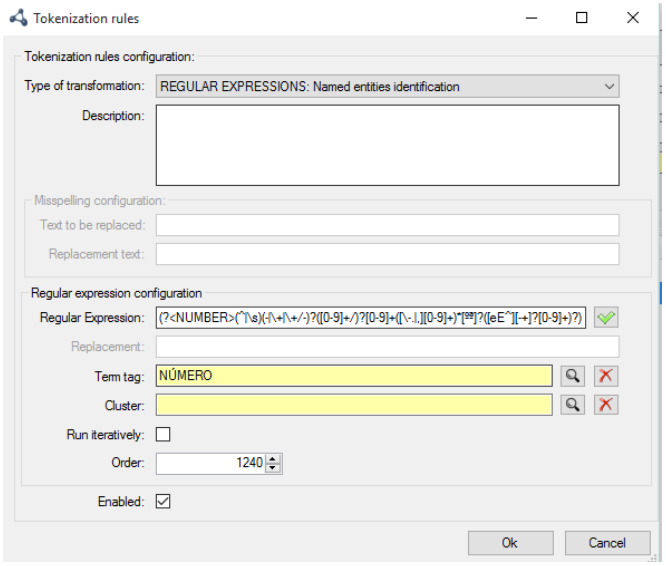

Ilustración 90 – Reglas de tokenización – Etiquetado de números

Con la aplicación de esta regla, el conjunto de caracteres 1234567 se etiqueta como un <NÚMERO>. De esta forma, cuando en una regla o patrón se encuentre el ítem <NÚMERO> se reconocerá cualquier secuencia de caracteres numéricos.

Esto que parece lógico en el tratamiento de expresiones basadas en el lenguaje natural, tiene consecuencias en la aplicación sobre expresiones que están fuera del lenguaje pero que se utilizan dentro del dominio de términos del proyecto.

La aplicación de la regla sobre el término del ejemplo: 100015005Antonio, provocaría la separación del término en dos términos, <NÚMERO> Antonio.

Con esta transformación se imposibilitaría el reconocimiento del perfil 100015005Antonio dentro del clúster de usuarios.

Al realizar un test sobre el término, podemos ver cómo se está aplicando la regla de tokenización que lo separa en dos partes. Marcado en rojo puede apreciarse cómo la aplicación de la regla anteriormente mostrada tiene para este caso un efecto no deseado.

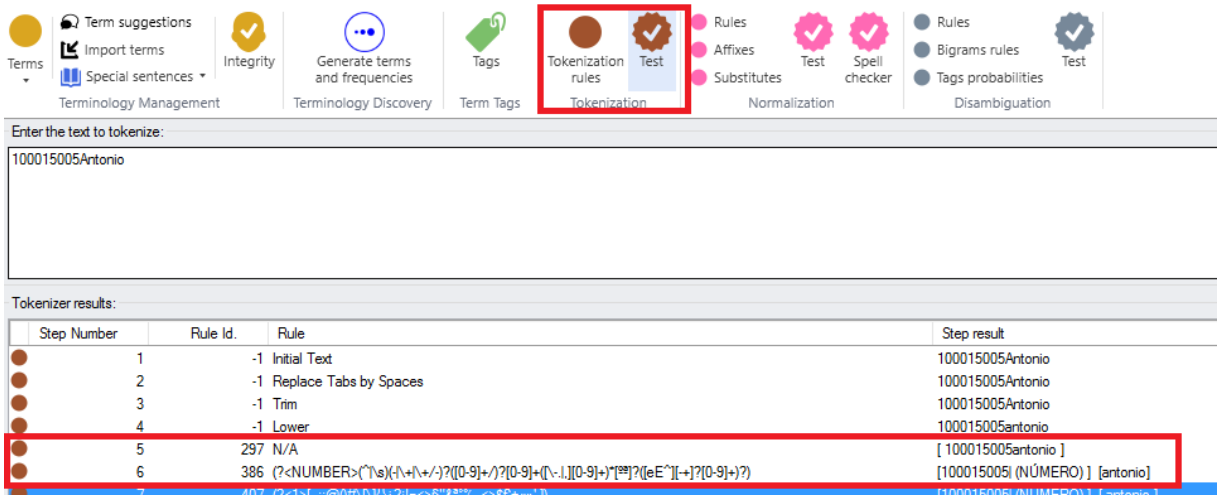

Ilustración 91 – Reglas de tokenización – Aplicación no deseada

Para la resolución de esta problemática se han encontrado dos posibles soluciones.

- Alteración de la regla.
  Si nos fijamos en la expresión regular sobre la que se aplica la regla de etiquetación, finaliza con un comodín.

  (?<NUMBER>(^|\s)(-|\+|\+/-)?([0-9]+/)?[0-9]+([\-.|,][0-9]+)*[ºª]?([eE^][-+]?[0-9]+)?)

  Esta definición es la que ha provocado el comportamiento no deseado para el caso que nos ocupa.

Si lo eliminamos, las expresiones del tipo 111111…111<CUALQUIER_COSA> no serán clasificadas como números.

En las siguientes ilustraciones podemos apreciar muy claramente el comportamiento en uno y otro caso.

- Aplicación de la regla de clasificación con expresión regular con comodín. El término es dividido clasificando la primera parte del mismo como un número.

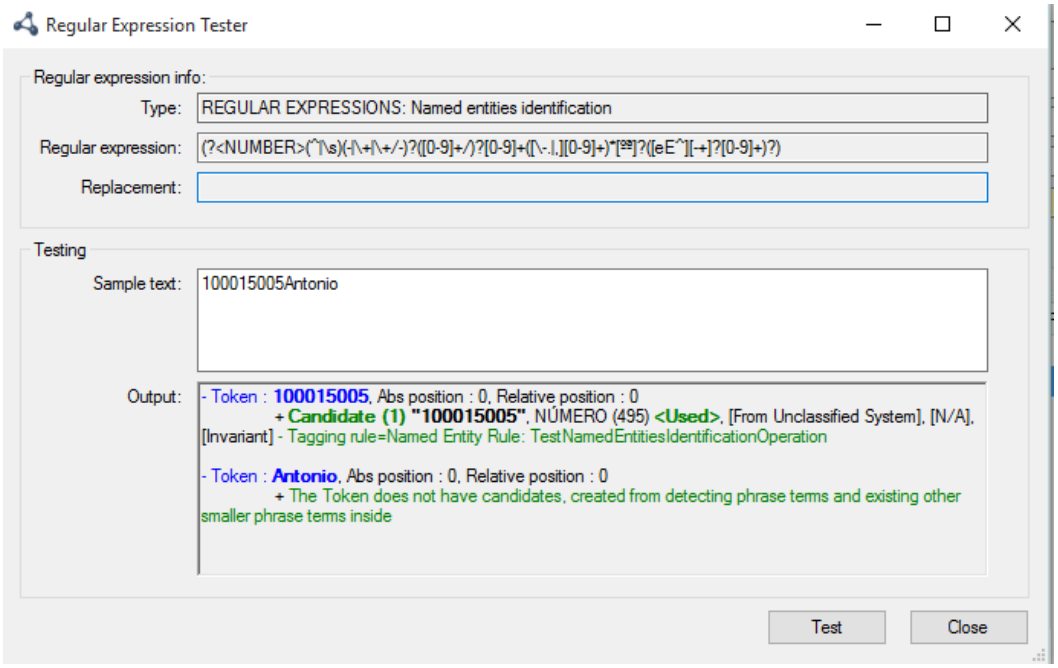

Ilustración 92 – Reglas de tokenización – Test expresión regular con comodín

- Aplicación de la regla de clasificación con expresión regular sin comodín. No se aplica ninguna transformación.

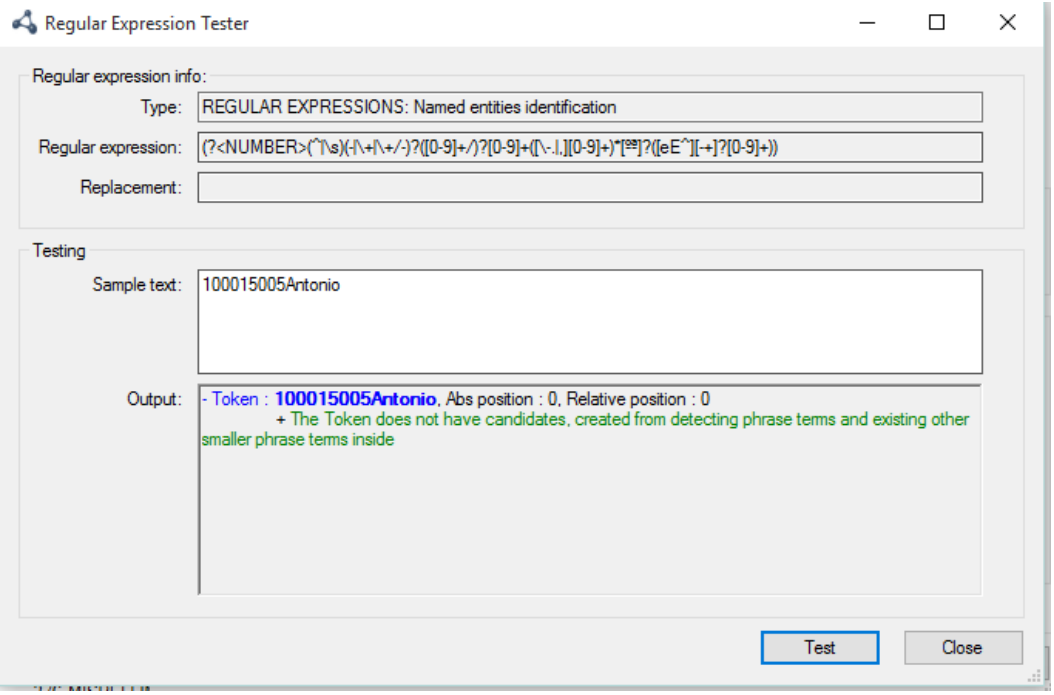

Ilustración 93 – Reglas de tokenización – Test expresión regular sin comodín

- Creación de reglas específicas para los términos problemáticos.

  La segunda solución que se puede aplicar para resolver el problema es definir reglas específicas para los términos que generan algún tipo de conflicto.
  Se puede por ejemplo crear una regla que convierta el término 100015005Antonio en Antonio100015005, de esta forma el término no será afectado por la catalogación de números.

Como puede observarse, dentro de la elaboración de la ontología y pese a los esfuerzos en la etapa de recopilación y transformación de la información, surgirán situaciones particulares que demandarán de un tratamiento a nivel de token.

En la problemática expuesta el problema se ha resuelto mediante el primer enfoque: modificar la regla *out-of-the-box* de la herramienta. El segundo enfoque es también factible dado que el conjunto de términos que plantean esta problemática dentro del dominio de datos a analizar es de sólo dos términos con lo que su implementación no es muy costosa, sin embargo el primer enfoque ofrece un mayor recorrido y versatilidad ante potenciales nuevos términos que se encuentren en este situación y además no ha afectado al tratamiento y reconocimiento posterior de la información objeto del estudio.

6.4. Patrones

Una vez alcanzado un punto óptimo de organización de la información en base al trabajo sobre las tareas anteriores, llegamos a la tarea de elaboración de las reglas y principios en los que se encuadran todos los *tweets* recogidos. Es decir, las sintaxis de los *tweets*.

La forma en la que se define la sintaxis para el estudio se enmarca dentro de la creación de patrones dentro de la herramienta *knowledgeMANAGER*.

Se ha seguido una aproximación empezando por la definición de un conjunto de patrones básicos cuya combinación nos lleva a patrones compuestos. La combinación de los patrones compuestos nos lleva a los patrones finales que son los patrones que representan los *tweets*.

- Patrones básicos.

  Distinguimos a su vez dos tipos de patrones.

  - Patrones básicos de estructura de *tweet.*
    El contenido del *tweet*, el mensaje en sí, difiere de un mensaje a otro, no en vano cada *tweet* es una opinión expresada por un alumno. Sin embargo hay una parte del *tweet* que se corresponde con la estructura propia del *tweet*.

    En el siguiente ejemplo, marcado en gris podemos apreciar la estructura básica del *tweet*.

"En el tweet: 605076621717053400, en respuesta al tweet: 573776088377262100, el usuario roxana10720373 comenta: miisiuc3m A mí me gusta más el tema de estrategia y transición de servicio, ya que son el punto inicial e intermedio de todo el ciclo."

    Existirá por lo tanto un conjunto de patrones que representen esta estructura. Los contenedores que se definen en el patrón se corresponden con la representación del *tweet* realizada en las fases de recolección y transformación/preparación de la información.

    - Usuario del tweet.
      Usuario que ha publicado el *tweet*.

    - Identificador del *tweet*.
      Número clave del *tweet*.

    - *Tweet* en respuesta de otro *tweet*.
      Si el *tweet* es una respuesta de otro *tweet*, se incluye información del *tweet* original sobre el que se produce la respuesta.

  - Patrones básicos de carácter del tweet.

    Entrando ya en el contenido del texto del *tweet*, se va a analizar el tipo del *tweet*, entendiendo por tipo del *tweet* la pertenencia a alguno los tipos definidos en el análisis.

    - Tweets positivos.
    - Tweets negativos.
    - Tweets de opinión.
    - Tweets de pregunta.
    - Estándares o certificaciones.
    - Sarcasmo.

- Patrones compuestos.

  Mediante la combinación de varios patrones básicos, se definen patrones compuestos. Representamos una regla sintáctica sencilla que puede ser combinada con otros elementos. Es de esta forma con la que se va creando una estructura más compleja de patrones.

  Sobre el ejemplo anterior, el patrón de definición global del *tweet* estará compuesta por tres patrones básicos: un patrón en el que se recoge la estructura de composición de la identificación del *tweet* (en amarillo), otro patrón en el que se identifica el usuario que publica el *tweet* (en azul) y finalmente otro patrón básico en el que se recoge (si el *tweet* es una respuesta) la identificación del *tweet* al que se responde (en verde).

  > *En el tweet: 605076621717053400, en respuesta al tweet: 573776088377262100, el usuario roxana10720373 comenta:*
  >
  > *En el tweet: 605076621717053400*
  >
  > *En respuesta al tweet: 573776088377262100*
  >
  > *El usuario roxana10720373 comenta*

  Continuando con el análisis de los patrones compuestos, además del patrón global de estructura del *tweet*, también se definen patrones compuestos con las diferentes alternativas de cada una de las características de *tweet* objeto del análisis (positivos, negativos, etc.).

  Aquí podemos ver dos *tweets* que habrán de ser englobados dentro de la categoría de positivos.

  > *"miisiuc3m me ha gustado la perspectiva práctica sobre ITIL de la charla de hoy"*
  >
  > *"miisiuc3m Muy buena presentación de ontologías. Gracias David!"*

  Cada uno de los *tweets* será representado por un patrón de *tweet* positivo diferente.

- Patrones finales.

  El patrón compuesto ya va agrupando las características individuales de cada parte del *tweet* así como de su estructura global.

  Es en el patrón final en el que se define la estructura global de los *tweets*.

  Por ejemplo la combinación del patrón que representa globalmente a *tweet* con el patrón que agrupa las diferentes estructuras de *tweets* positivos dará lugar al patrón final de *tweet* positivo.

- Patrones complejos.

  Una vez finalizado el trabajo de creación de patrones y en pleno proceso de prueba de cada uno de los *tweets* dentro de la herramienta, nos encontramos con que existen algunos *tweets* que expresan dos condiciones relevantes para el estudio.

  Es en este momento en el que surgen los patrones complejos. Son patrones que engloban *tweets* que expresan dos características a la vez. Por ejemplo un *tweet* positivo sobre estándares, o un *tweet* negativo sobre certificaciones.

6.4.1 Patrones básicos

La estructura de definición del *tweet* fue definida durante la fase de transformación de la información.
La utilidad transformTwitterData transforma la representación *json* del objeto *tweet* extraída del API REST de Twitter en una frase del tipo:

En el tweet: <tweetID>, en respuesta al tweet: <tweetID>, el usuario <usuario> comenta: <texto del tweet>

La parte marcada en gris vendrá representada por los patrones básicos de estructura del *tweet* mientras que la parte roja será el objeto de los patrones básicos de carácter del *tweet*.

- Patrones básicos de estructura del *tweet*.

  Dado que esta parte de la expresión es construida de forma constante por la herramienta de transformación, los patrones que surgen de su análisis son bastante sencillos y acotados.

  - Patrón de Identificación del *tweet.*

    Todos los patrones creados en knowledgeMANAGER durante el presente trabajo se han nombrado como: (PFC MINERÍA SENTIMIENTOS) - Nombre del patrón.

    En este patrón la parte variable es en la que se incluyen todos los identificadores de *tweet* objeto del estudio.

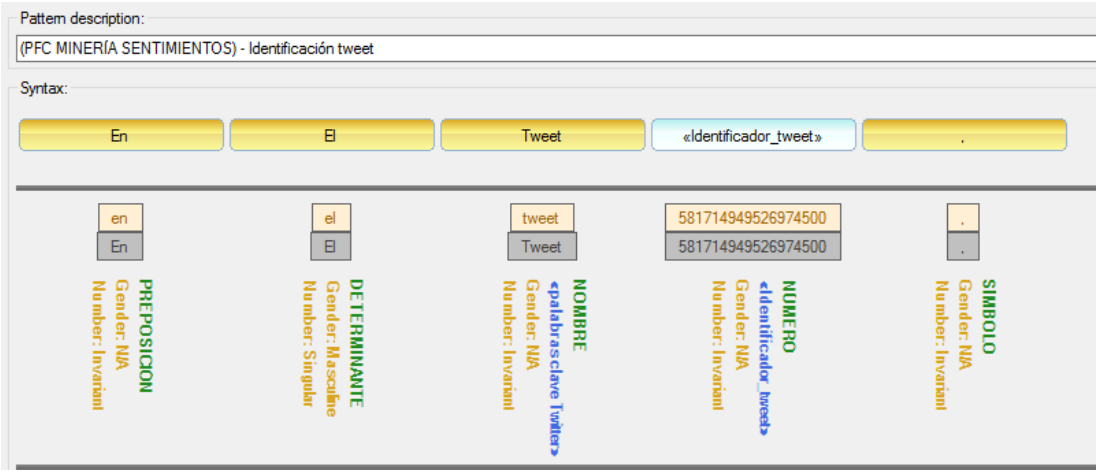

Ilustración 94 – Patrones – Identificación tweet

El slot <<Identificador_tweet>> se instancia con el clúster que contiene todos términos con los identificadores de los *tweets*.

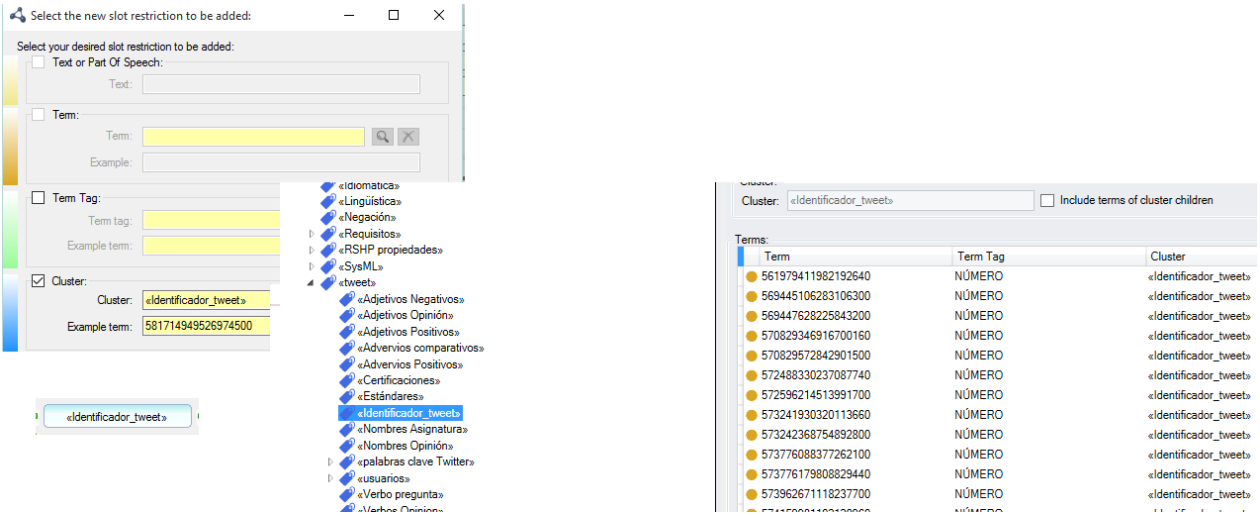

Ilustración 95 – Patrones – Identificación tweet – Cluster Identificador_tweet

- Patrón de *tweet* de respuesta.

  Es muy similar al anterior, la estructura de la frase es diferente, pero vuelve a tener un slot que se instanciará con los identificadores de *tweets* referenciados en el clúster correspondiente.

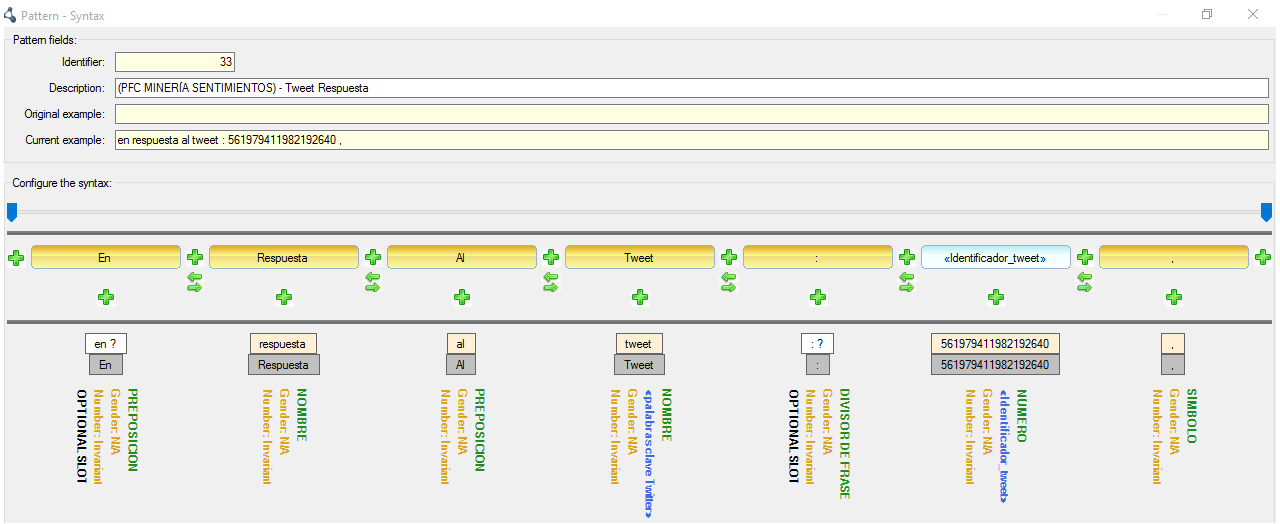

Ilustración 96 – Patrones – Tweet respuesta

- Patrón de usuario que publica el *tweet*.

  Nuevamente es un patrón con gran parte de la estructura definida de manera fija. La novedad en este caso es el slot de usuarios que se instancia con alguno de los valores del correspondiente clúster de usuarios.

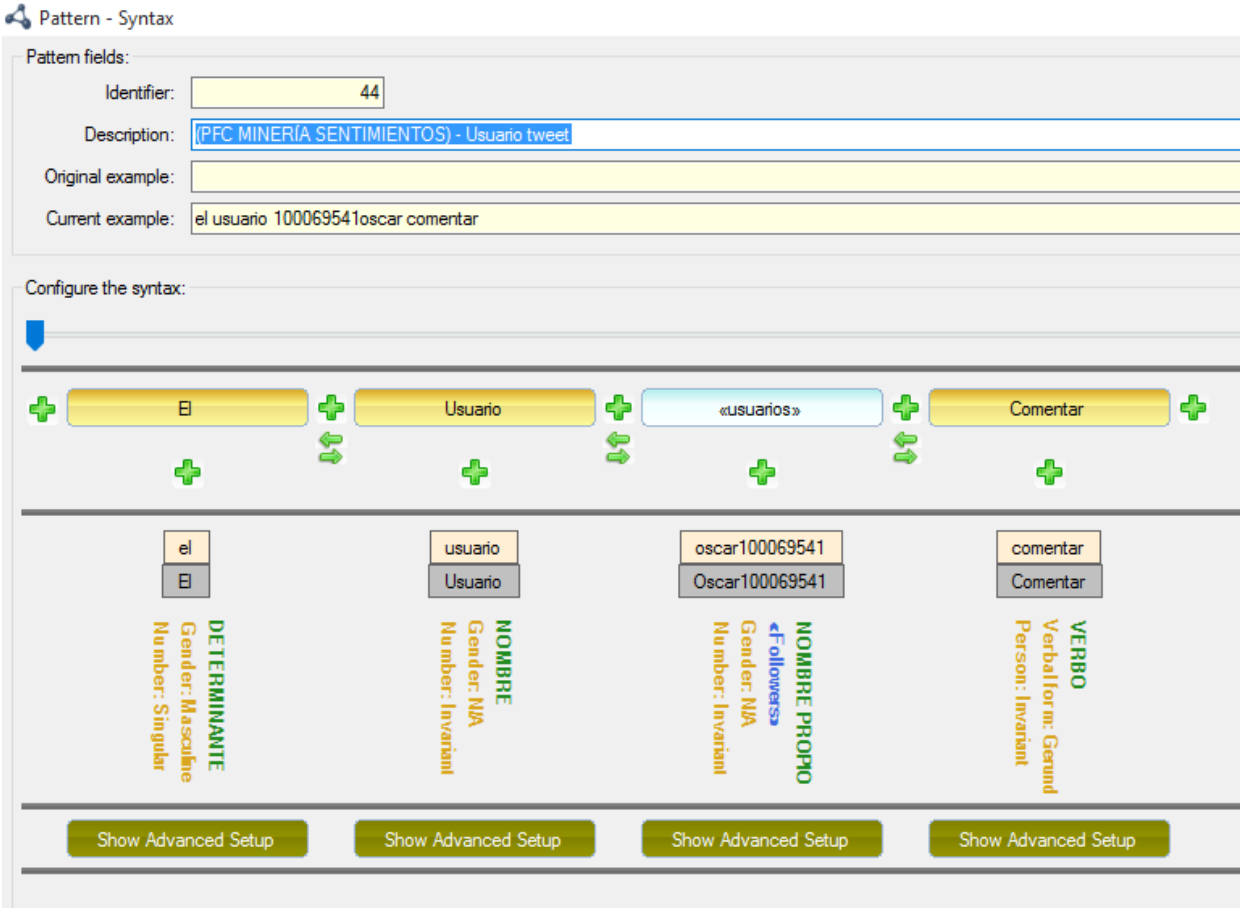

Ilustración 97 – Patrones – Usuario que publica el tweet

- Patrones básicos de carácter del *tweet*.

  Entramos en esta parte a crear los patrones sobre el texto libre del mensaje. En primer lugar se ha procedido con la abstracción de todos los mensajes, obviando la parte de estructura del *tweet*.
  Para facilitar esta labor y no tener que extraerlos manualmente, se sobrecargaron las operaciones de extracción de *tweets* de herramienta de transformación de datos.
  Mediante la marca con un simple *checkBox* la herramienta muestra el *tweet* completo o sólo el texto del mismo.

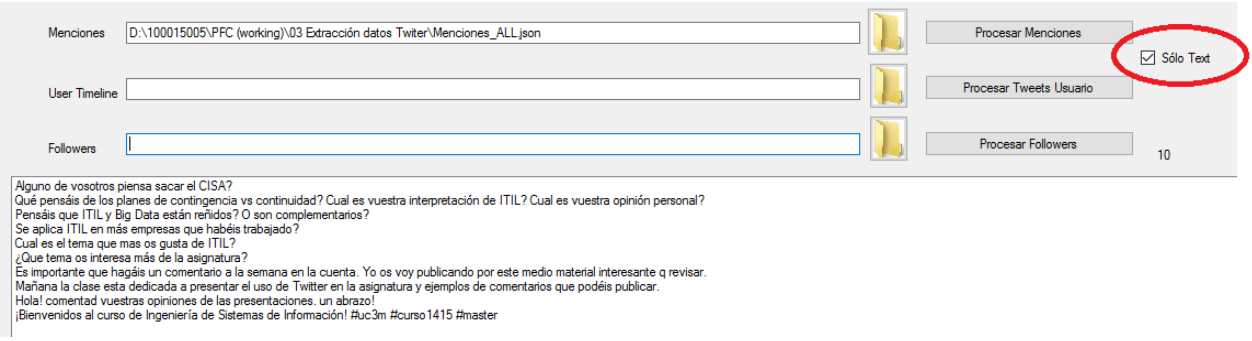

Ilustración 98 – Extracción de sólo el texto del tweet

El siguiente paso consiste en analizar cada uno de los *tweets*, clasificarlos como: Positivos, Negativos, Opinión, Pregunta, Sarcasmo o Estándar/Certificación, e indicar las palabras clave de la expresión que nos llevan a esa conclusión.

La siguiente tabla muestra un fragmento de la clasificación realizada para *tweets* positivos.
Este proceso se ha seguido para todos y cada uno de los *tweets*, quedando clasificado cada uno en base a lo que el autor ha querido expresar.

TEXTO	PALABRAS CLAVE
miisiuc3m gracias a tods por la participación en la última clase, debates entretenidos y una presentación bien preparada!	BIEN PREPARADA
miisiuc3m Una clase entretenida con dos debates para despedir el curso. Buen Fin de Semana!	CLASE ENTRETENIDA
miisiuc3m Muy buena presentación de ontologías. Gracias David!	MUY BUENA
miisiuc3m completa la presentación de CMMI. No lo conocía. Muy interesante.	MUY INTERESANTE
miisiuc3m hoy los nuevos técnicos han tenido el aula configurada a tiempo y hemos empezado la clase súper puntuales #pmiisuc3m FELIZ!	#pmiisuc3m, FELIZ!
miisiuc3m Muy interesante la ponencia de Itil por parte de Emiliano!	MUY INTERES*ANTE*
miisiuc3m gracias FELIZ!	FELIZ!

Tabla 39 – Extracto de clasificación tweets positivos

Finalmente, las palabras o expresión clave, van a ser las que provoquen la definición del patrón o patrones que engloben cada una de las posibilidades expresadas.

Se muestran a continuación ejemplos de un patrón de cada una de las categorías.

- Positivos.

  En este punto el hecho de que la herramienta esté precargada con la gramática castellana es de inestimable ayuda ya que evita crearse etiquetas o clústeres de elementos comunes

en lengua castellana. Existe precargada toda la información relativa a determinantes, artículos, adverbios, verbos y nombre comunes, adjetivos, etc.

Podrá observarse en este patrón y en otros patrones básicos cómo se combina el uso de elementos ya disponibles, en este caso el ADVERVIO con elementos creados ad-hoc para este proyecto como el clúster de <<Adjetivos Positivos>>.

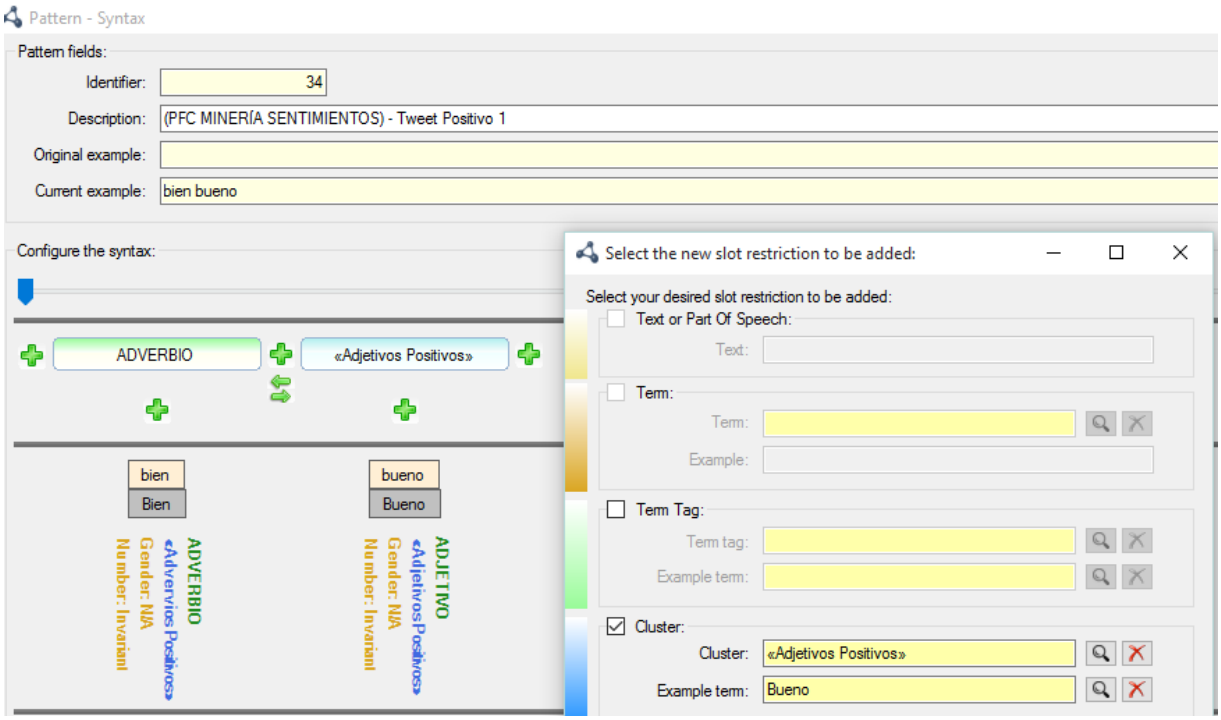

Ilustración 99 – Patrones – Patrón básico positivo

- Negativos.

En este caso el patrón es bastante sencillo ya que los adjetivos negativos encontrados siempre aparecen claramente relacionados con una expresión negativa.

No era el caso de las expresiones positivas donde por ejemplo el token “bueno” puede ser un adjetivo o también un adverbio o formar parte de una expresión coloquial:

“clase muy buena”
“bueno, se ha acabado la clase y quedan cuestiones pendientes”

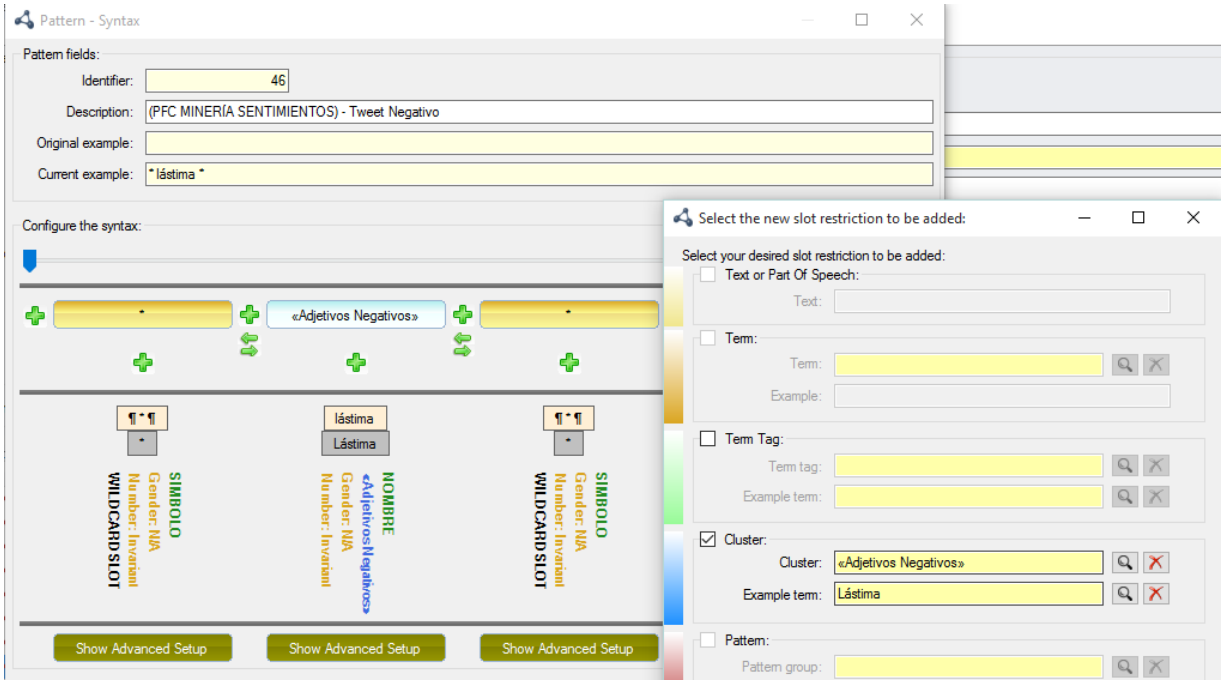

Ilustración 100 – Patrones – Patrón básico negativo

- Opinión.

  Sin duda esta es la categorización que ha resultado más complicada. Una opinión puede expresarse con múltiples combinaciones, incluso incluyendo adjetivos positivos o negativos en la propia opinión o comparación.
  Es por ello que sea la categoría que más patrones haya provocado (5) conjuntamente con los positivos, en este caso por volumen (6).

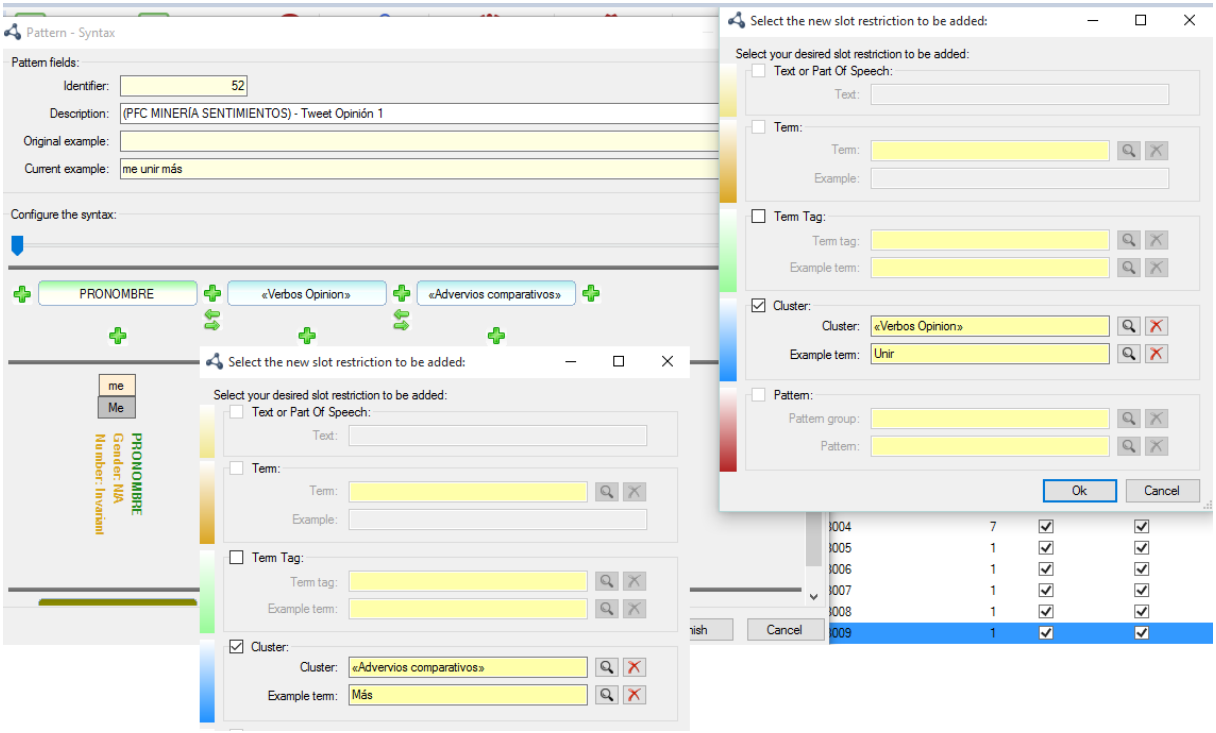

Ilustración 101 – Patrones – Patrón básico opinión

En este patrón podrán englobarse los siguientes *tweets*.

TEXTO	PALABRAS CLAVE
miisiuc3m me gusta mucho la imagen, vale mas que mil palabras...	ME GUSTA MUCHO
miisiuc3m A mi me gusta más el tema de estrategia y transición de servicio, ya que son el punto inicial e intermedio de todo el ciclo.	A MÍ ME GUSTA MÁS
OscarSipele miisiuc3m efectivamente me uno al comentario de Óscar	ME UNO AL COMENTARIO
miisiuc3m el esquema del primer slide me parece abrumador	ME PARECE

Tabla 40 – tweets opinión del patrón básico opinión 1

- Pregunta.

  En este caso nos encontramos con dos patrones.
  Generalmente todas las expresiones que denotan una pregunta contienen el símbolo de cierre de interrogación: “?”.

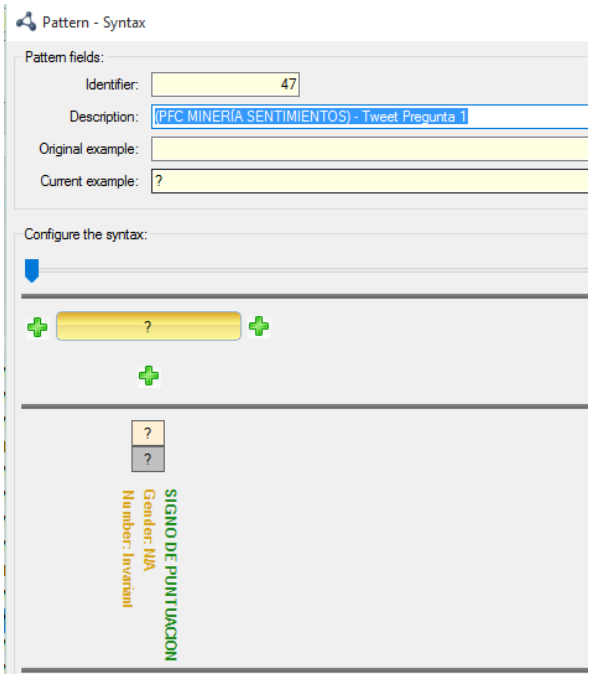

Ilustración 102 – Patrones – Patrón básico pregunta 1

Existe una excepción. Se ha encontrado un caso que denota interrogación sin símbolo de interrogante, en él se utiliza la acción PREGUNTAR, por lo que se ha creado otro patrón para recoger este caso y que potencialmente recogería todos los *tweets* que pudiesen encontrarse en este circunstancia en un futuro.

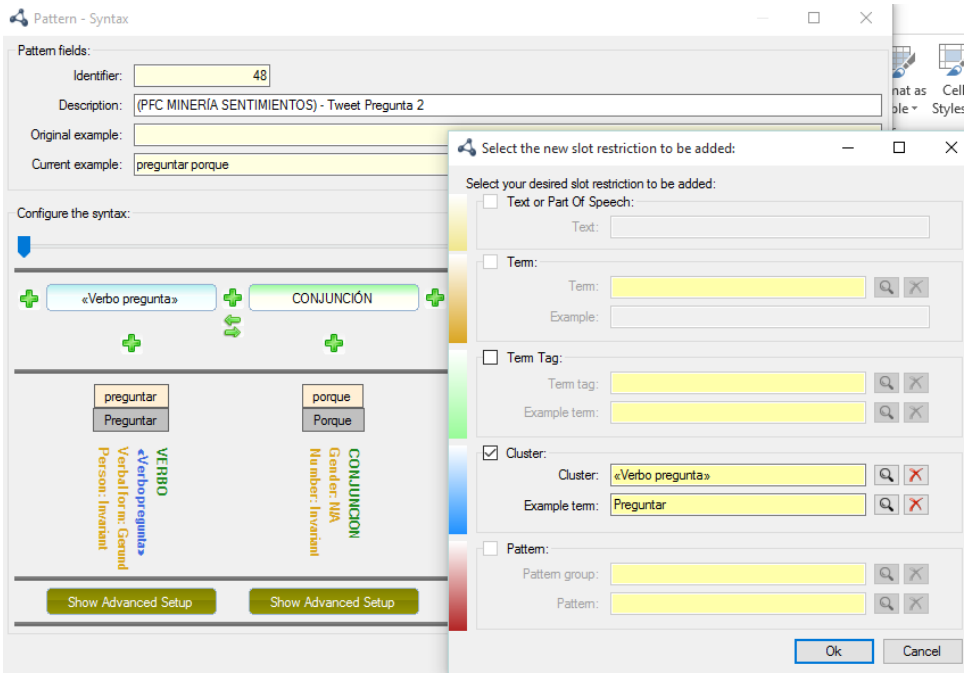

Ilustración 103 – Patrones – Patrón básico pregunta 2

- Sarcasmo.

  El patrón definido para esta caso se basa únicamente en el *hashtag* definido "*#smiisuc3m".*

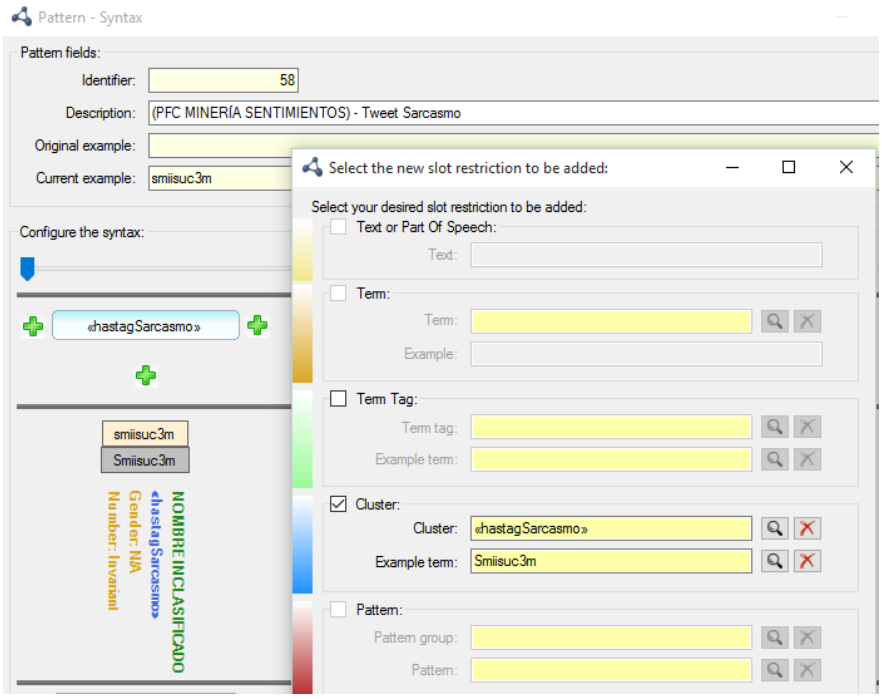

Ilustración 104 – Patrones – Patrón básico sarcasmo

- Estándar o certificación.

  También resulta objeto de estudio el número de menciones que se hacen a certificaciones o estándares de la industria y sobre los que ha versado parte del contenido de la asignatura.

  Como novedad con respecto a los patrones anteriores, en este patrón se utiliza el operador *OR* para definir la posibilidad de que un *token* sea del clúster de <<Estándares>> o <<Certificaciones>>.

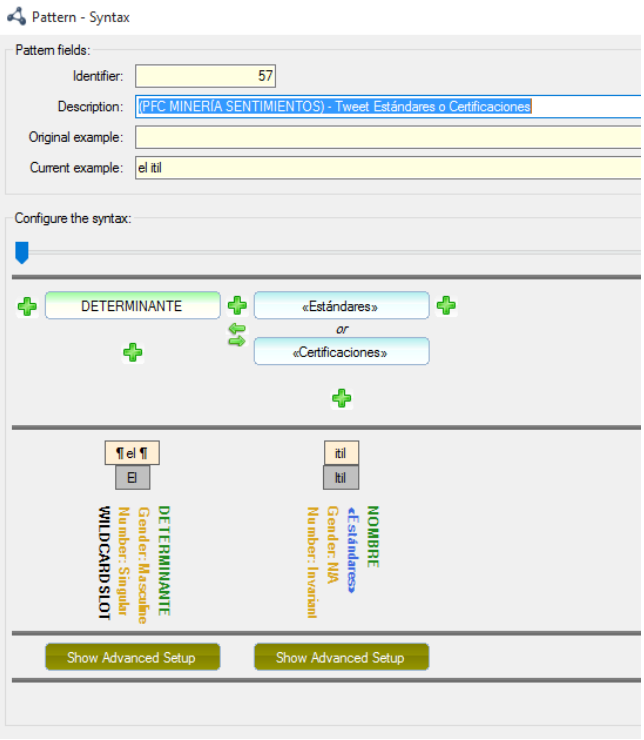

Ilustración 105 – Patrones – Patrón básico estándares o certificaciones

6.4.2 Patrones compuestos

Sobre los patrones básicos se produce una asociación por semántica, de forma que aquellas clasificaciones que tienen varios patrones básicos, se agrupan en un solo patrón para facilitar su posterior tratamiento.

Es el caso de los patrones positivos, de respuesta y de opinión.

Se han creado seis patrones básicos de carácter positivo. Un patrón compuesto, define un *tweet* como positivo cuando el texto del *tweet* se corresponde con alguno de los seis patrones individuales definidos.

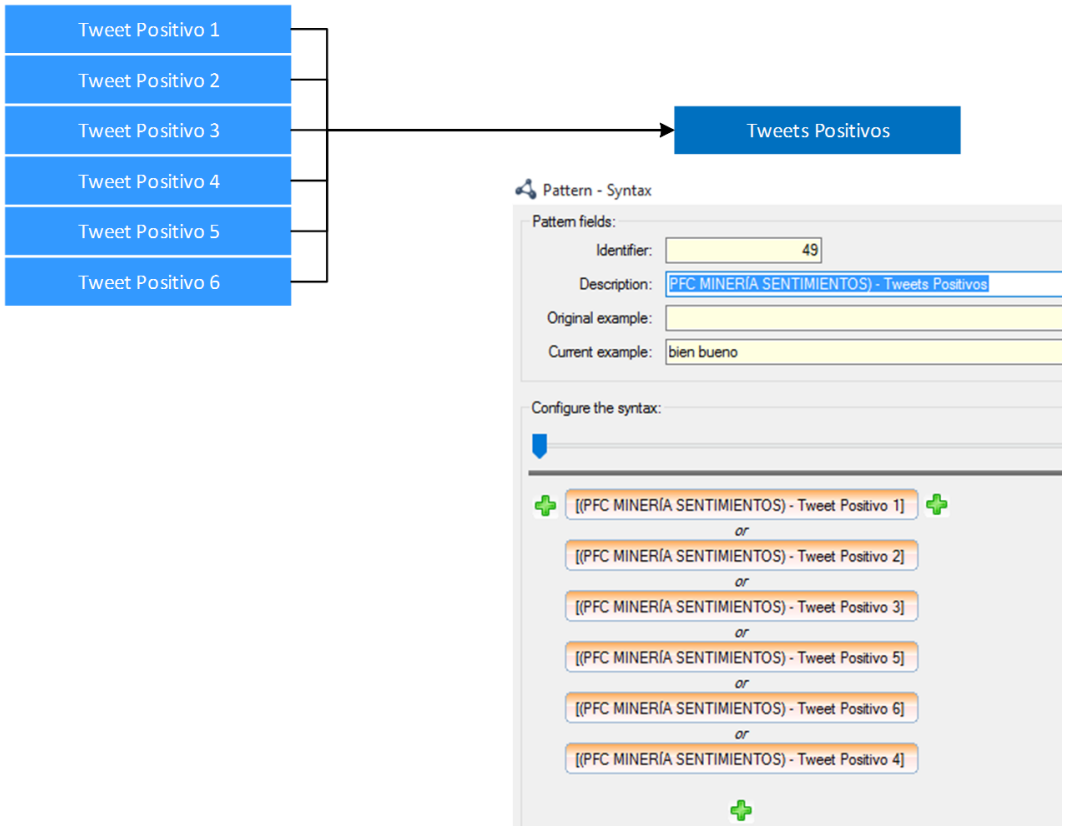

Ilustración 106 – Patrones – Patrón compuesto positivo

El mismo caso se da para los *tweets* de Opinión y Pregunta donde nuevamente se define un patrón que agrupa con cláusula OR los diferentes patrones básicos.

El resto de clasificaciones no ha exigido de una agrupación ya que tienen un único patrón para representar sus *tweets*. Estamos hablando de Negativos, sarcasmo y estándar/certificación.

Los patrones compuestos tratados hasta el momento persiguen el objetivo de agrupar todas las posibilidades en un único patrón, simplificando la definición de patrones más complejos, en los que se trate el *tweet* globalmente.
Existe sin embargo otro patrón compuesto que no se basa en alternativa de patrones básicos, sino que aplica el operador AND sobre patrones básicos.
Se trata del patrón que define la estructura global de un *tweet*.

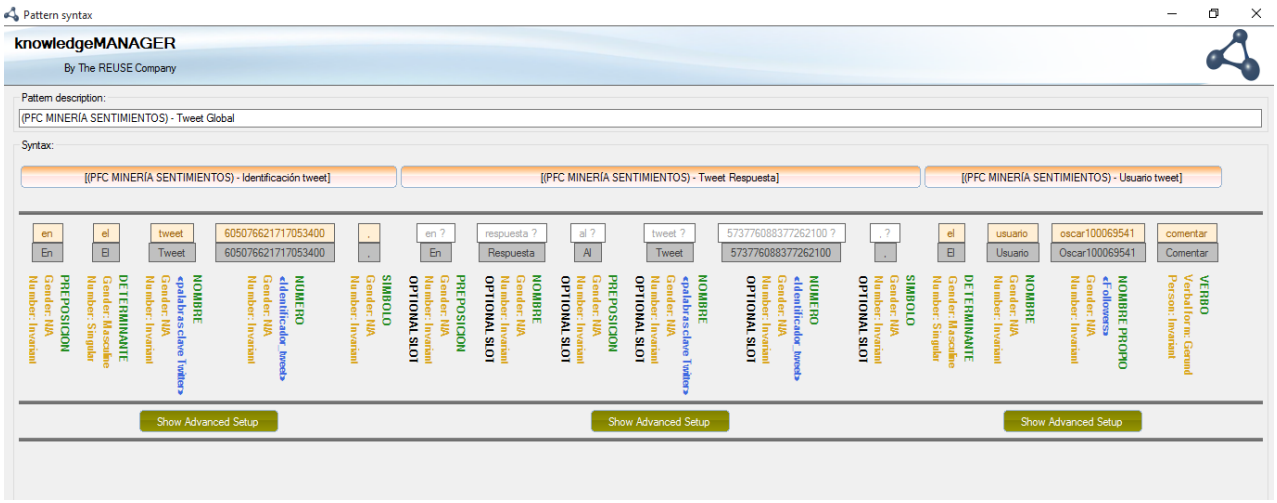

Ilustración 107 – Patrones – Patrón compuesto definición global

6.4.3 Patrones finales

Los patrones finales recogen la sintaxis completa de reconocimiento de un *tweet*.

La creación del patrón final se realiza mediante la combinación de los patrones básicos que tenían entidad suficiente por sí mismos para no formar parte de un patrón compuesto y los patrones compuestos surgidos de la interpretación conjunta de varios patrones básicos.

Existe un elemento que es indispensable en todos los patrones y es el patrón compuesto de definición global, este patrón representa la definición del *tweet*, a partir de él se construyen los diferentes tipos de *tweets*. En el siguiente diagrama pueden apreciarse las relaciones que se han implementado hasta llegar al patrón final.

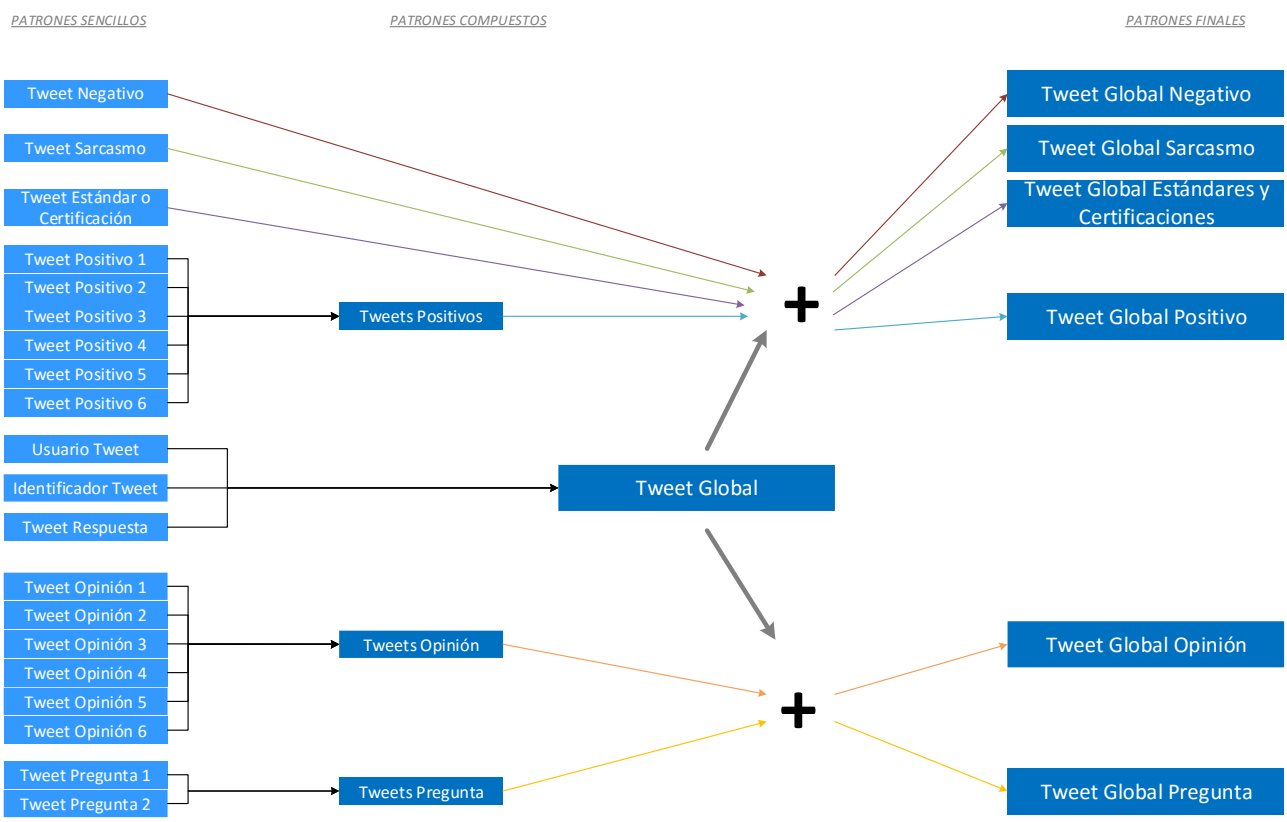

Ilustración 108 – Patrones – Patrón compuesto definición global

Existe por lo tanto seis patrones globales.

- Tweet Global Positivo.

  Compuesto por la definición global del *tweet* y la agrupación de todos los patrones que representan *tweets* positivos.

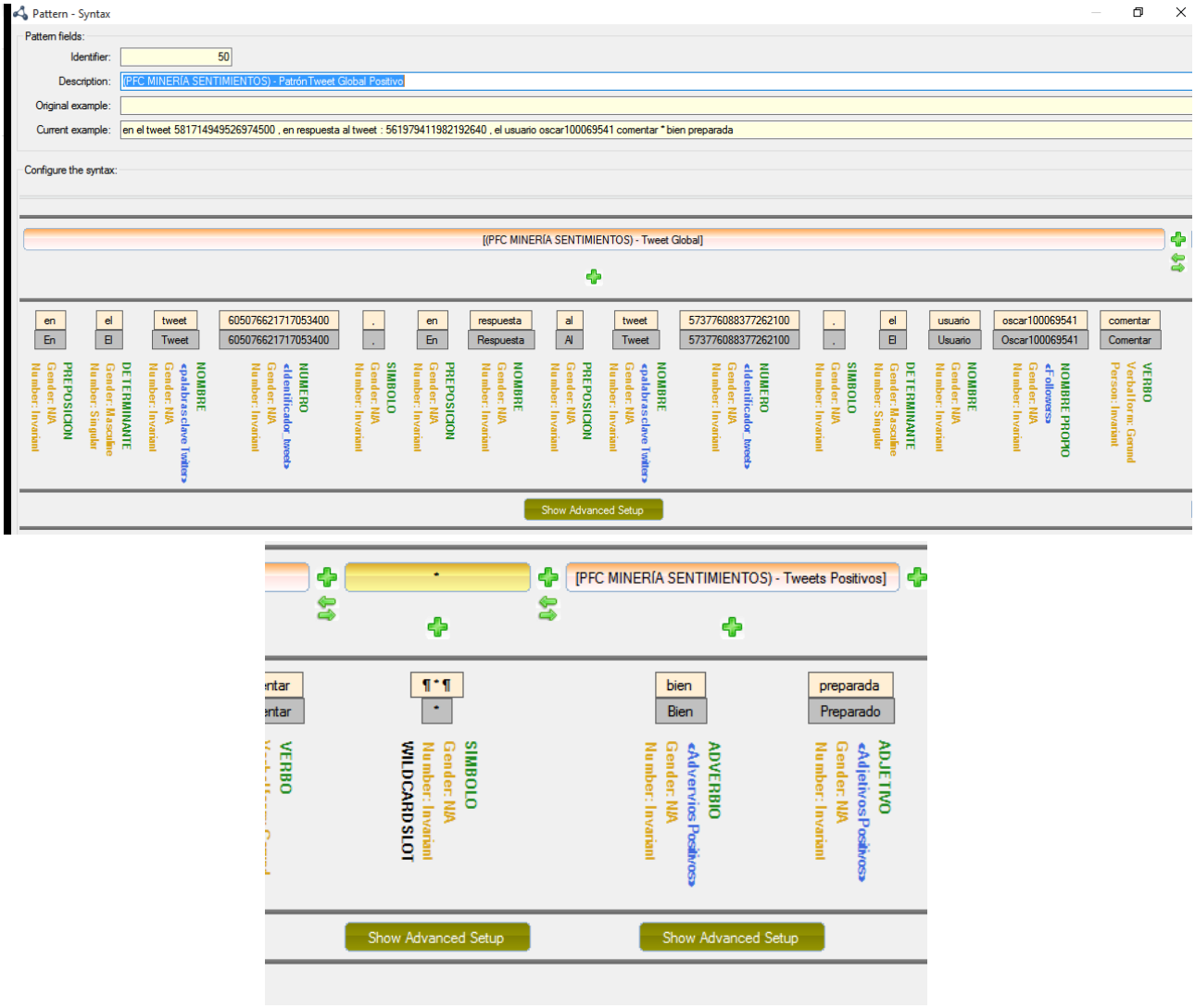

Ilustración 109 – Patrones – Patrón Global Positivo

Analicemos el comportamiento del análisis del patrón para un *tweet* positivo.

“

En el tweet: 601882814238347300, el usuario 100069541Oscar comenta: miisiuc3m interesante debate las posturas de cada persona nos acercan más al entendimiento de los temas propuestos.

“

Resulta interesante apreciar cómo el análisis va reconociendo las diferentes partes del *tweet* para ajustarlas al patrón que aplica en cada caso componiendo finalmente el patrón global positivo.

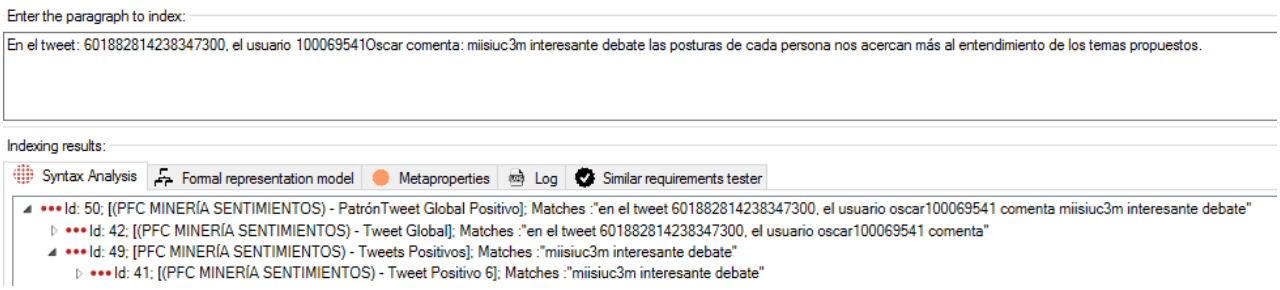

Ilustración 110 – Patrones – Análisis tweet positivos

- Tweet Global Negativo.

  Compuesto por el patrón global de definición de *tweet* y el patrón de *tweet* negativo.
  No existe patrón compuesto con los patrones negativos ya que sólo existe un patrón negativo.

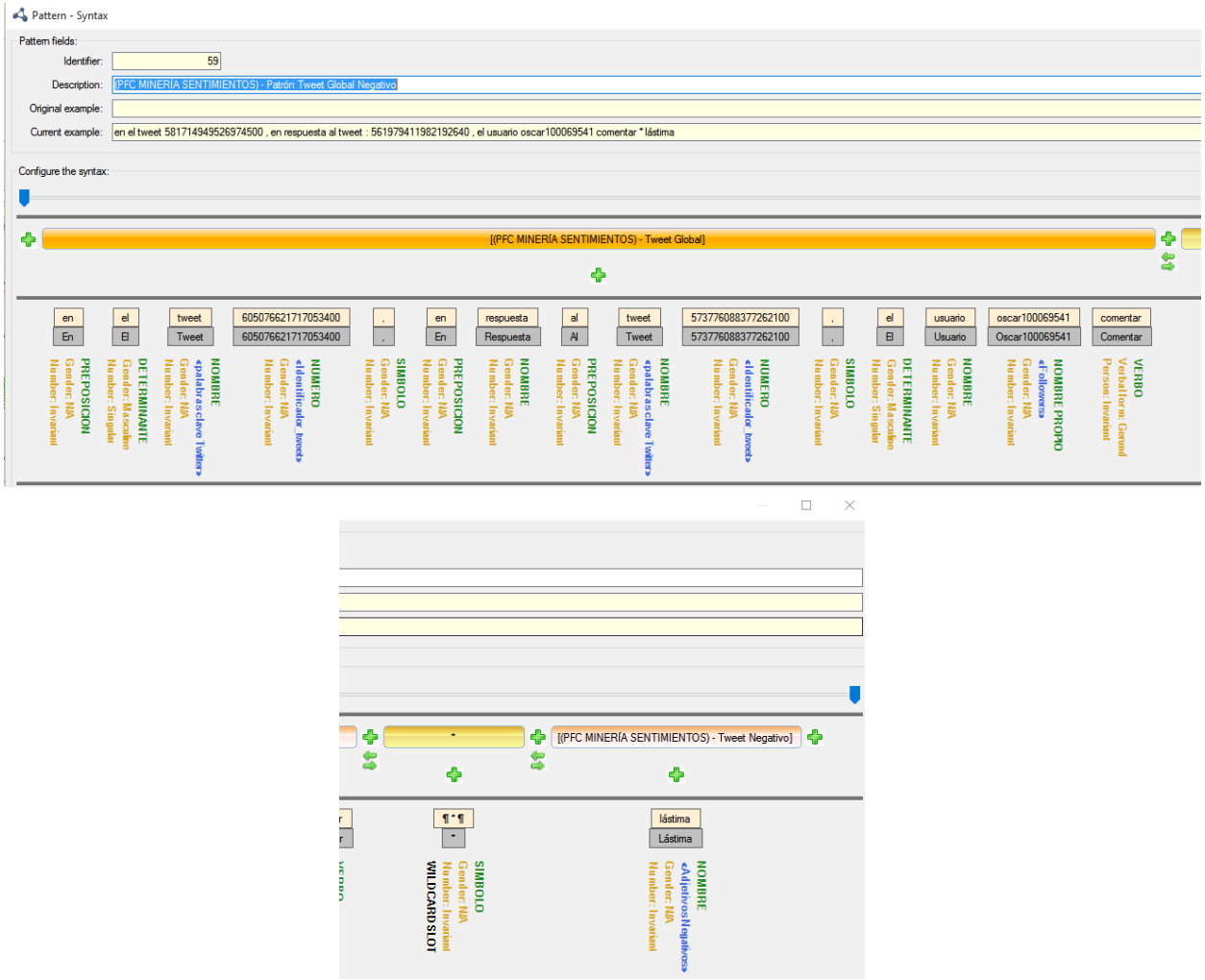

Ilustración 111 – Patrones – Patrón Global Negativo

El análisis de este tipo de *tweets*, negativos, es similar al análisis visto para los *tweets* positivos. Pero aprovecharemos el análisis de un *tweet* negativo para explicar la forma en la que se resuelve la ambigüedad de un *tweet*.
Procedamos con el análisis del siguiente *tweet*.

"En el tweet: 581561754993487900, el usuario eladiotercios comenta: miisiuc3m Buen resumen de la asignatura de auditoría por robertotdi. Lástima que se haya visto afectada por los problemas técnicos"

El *tweet* podría considerarse positivo y negativo a la vez. Nos encontramos ante un mensaje en el que el usuario expresa su satisfacción por el resumen de la asignatura pero puntualiza con el hecho de que ha habido problemas técnicos.

Cuando se aplican patrones sobre expresiones, hay ocasiones en las que pueden existir varios patrones que apliquen, es el momento de dar pesos a los patrones. En el caso de que se puedan aplicar dos patrones sobre una misma expresión, se aplicará aquel que tenga un peso mayor.

En la configuración mostrada en la imagen, el patrón de *tweets* negativos tiene un paso mayor (10010) que el patrón de *tweets* positivos (10004).

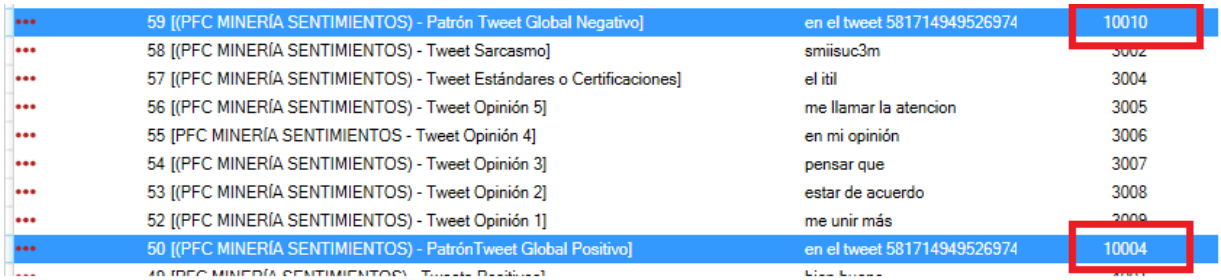

Ilustración 112 – Patrones – Aplicación patrón negativo por peso

Como consecuencia, el análisis de la sentencia, se ajusta a una sentencia de *tweet* negativo.

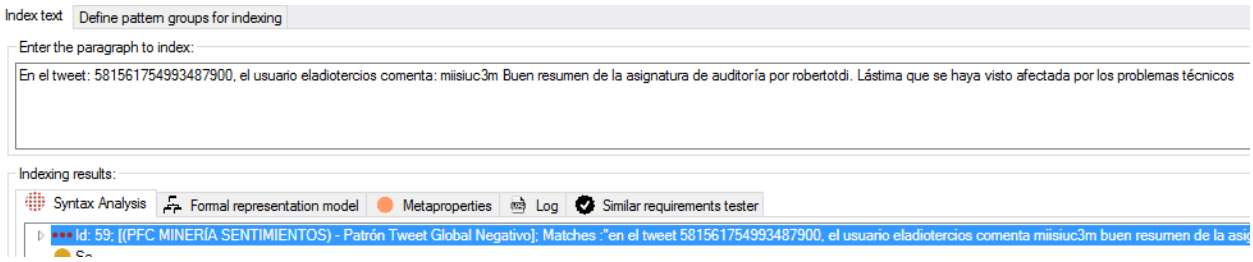

Ilustración 113 – Patrones – Análisis tweet negativo por pesos

Si por el contrario entendemos que el *tweet* positivo ha de tener más peso que el negativo, el patrón tendrá un peso superior (10004 sobre 10000).

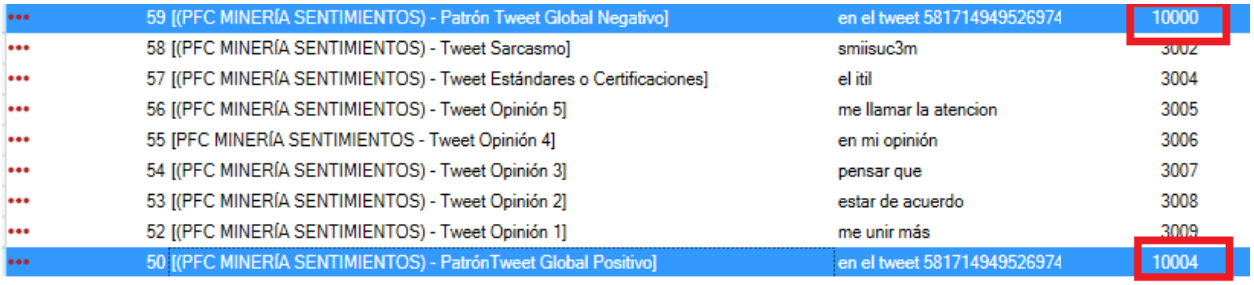

Ilustración 114 – Patrones – Aplicación patrón positivo por peso

De esta forma, en el momento del análisis de la expresión, se aplicará el patrón positivo.

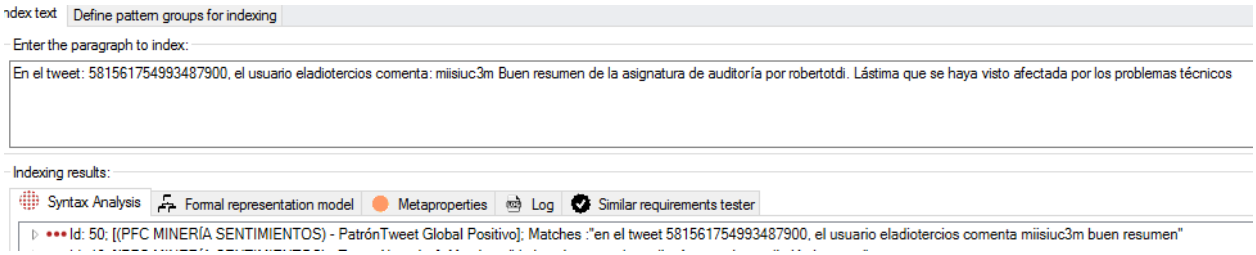

Ilustración 115 – Patrones – Análisis tweet positivo por pesos

- Tweet Global Pregunta.
- Tweet Global Estándares y Certificaciones.
- Tweet Global Sarcasmo.

La generación de patrones finales para estos tres tipos: pregunta, estándares/certificaciones/sarcasmo es similar a los anteriores. Una vez realizados los patrones básicos (sarcasmos tiene sólo un patrón) y compuestos para aquellas categorías con varios patrones para un mismo tipo (en este caso pregunta y estándares/certificaciones), la elaboración del patrón global es equivalente a las anteriormente descritas.

6.4.4 Patrones complejos

Hasta el momento hemos asumido que cada *tweet* es catalogado de una única forma, es decir, un *tweet* o es positivo, o es negativo, o es de certificaciones/estándares, etc.

En el caso en el que nos hemos encontrado con ambigüedad a la hora de calificarlo, hemos actuado dando pesos a los patrones de forma que se prioricen unos sobre otros. Este ha sido el caso del ejemplo visto con un *tweet* que según interpretaciones puede ser positivo o negativo.

Nos encontramos ahora con el caso de *tweets* que han de ser categorizados dentro de dos patrones.

“En el tweet: 577781887281803300, en respuesta al tweet: 577670597263622100, el usuario robertotdi comenta: miisiuc3m Creo que ITIL tiene una visión del plan de continuidad para establecer procedimientos preventivos y reactivos.”

En este primer ejemplo estamos ante un *tweet* que por un lado expresa una opinión: “Creo que ITIL tiene una visión…”. Pero además se está tratando sobre un estándar/certificación al hablar de ITIL.

“En el tweet: 596727722426114000, el usuario InfinitaMemoria comenta: miisiuc3m completa la presentación de CMMI. No lo conocía. Muy interesante.“

En este segundo ejemplo nos encontramos ante un *tweet* que podríamos calificar dentro de la categoría de estándares/certificaciones al hablar de CMMI y a su vez es un *tweet* de carácter positivo.

En estos casos no estamos hablando de aplicar un patrón u otro, sino que estamos hablando de combinar nuevamente patrones, en este caso combinar patrones finales para dar lugar a un nuevo patrón. Patrón que hemos denominado complejos.

La política que se ha seguido para la generación de los patrones complejos es la siguiente.

En primer lugar, dado que se ha detectado que estos patrones afectan a *tweets* de temática estándar/certificación, se ha combinado el patrón compuesto de estándar/certificación con los patrones compuestos de los tipos afectados: positivos, opinión y pregunta.

Sobre el patrón resultante, se ha añadido el patrón compuesto de definición global del *tweet* y el resultado es el patrón complejo.

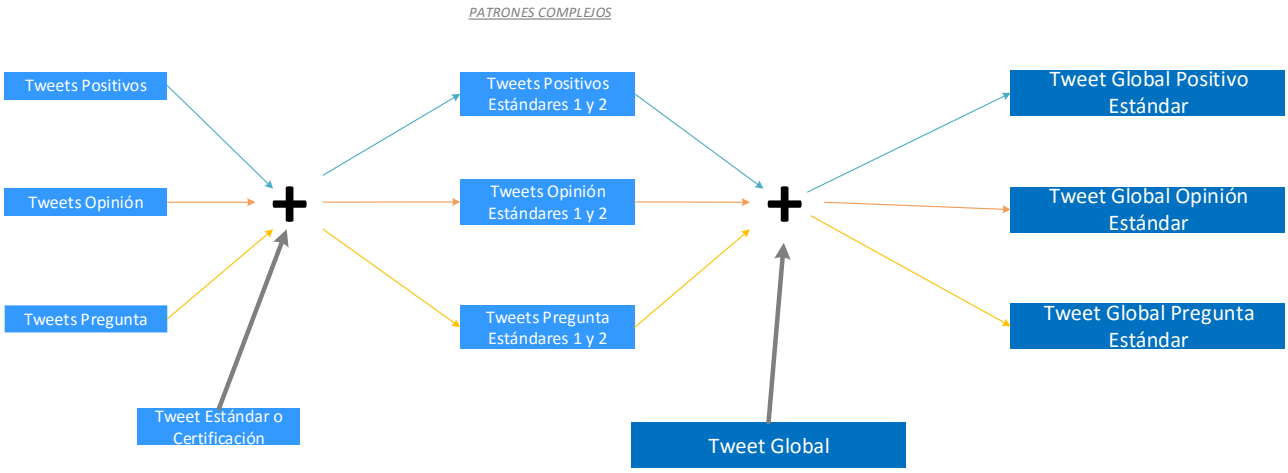

Ilustración 116 – Patrones – Patrones complejos

Continuando con los ejemplos anteriores.

- Patrón complejo opinión – estándar/certificación.

“En el tweet: 577781887281803300, en respuesta al tweet: 577670597263622100, el usuario robertotdi comenta: miisiuc3m Creo que ITIL tiene una visión del plan de continuidad para establecer procedimientos preventivos y reactivos.”

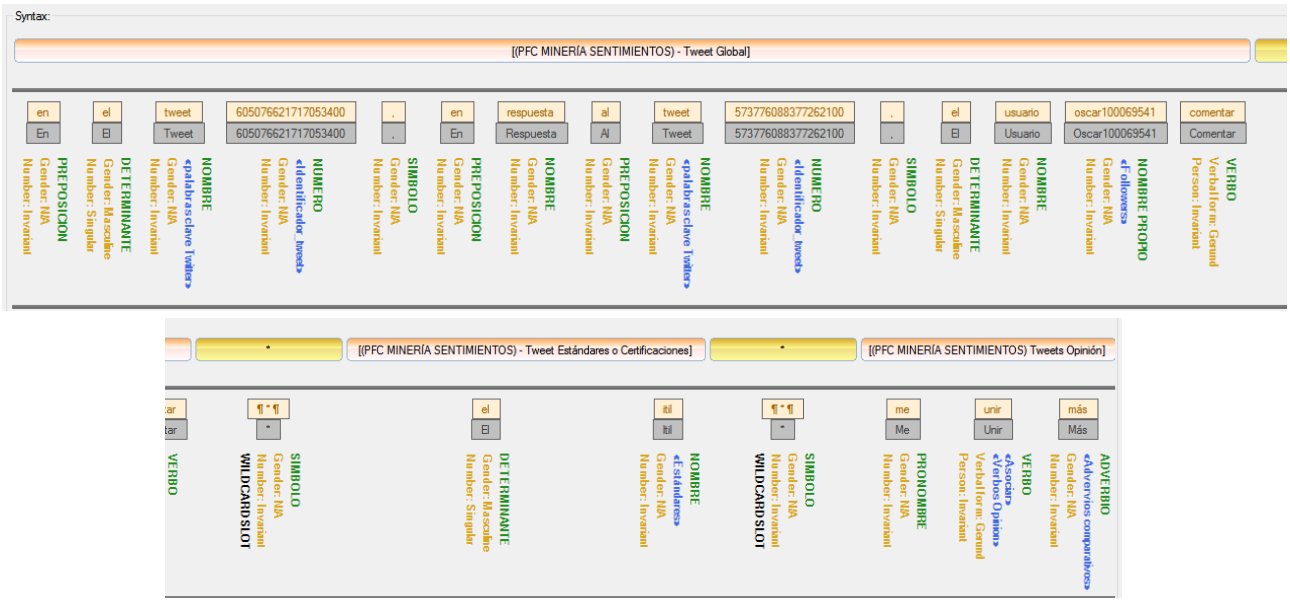

Ilustración 117 – Patrones – Patrón complejo opinión – estándar/certificación

El resultado de la ejecución para el *tweet*, es la aplicación del nuevo patrón. Patrón global de opinión sobre estándares/certificaciones.

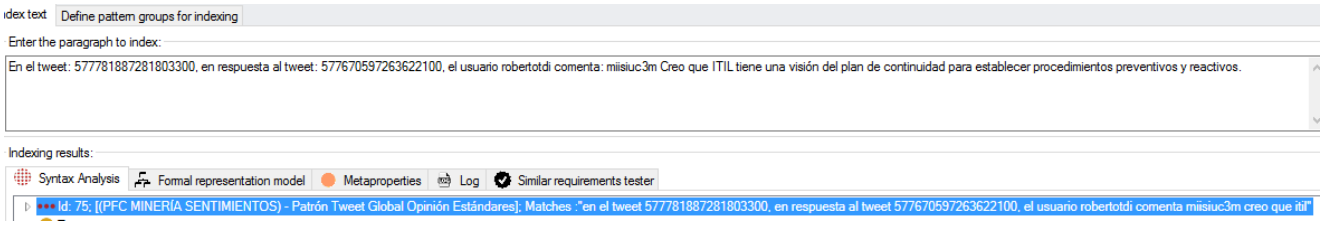

Ilustración 118 – Patrones – Aplicación patrón complejo opinión – estándar/certificación

- Patrón complejo positivo – estándar/certificación.

"En el tweet: 596727722426114000, el usuario InfinitaMemoria comenta: miisiuc3m completa la presentación de CMMI. No lo conocía. Muy interesante."

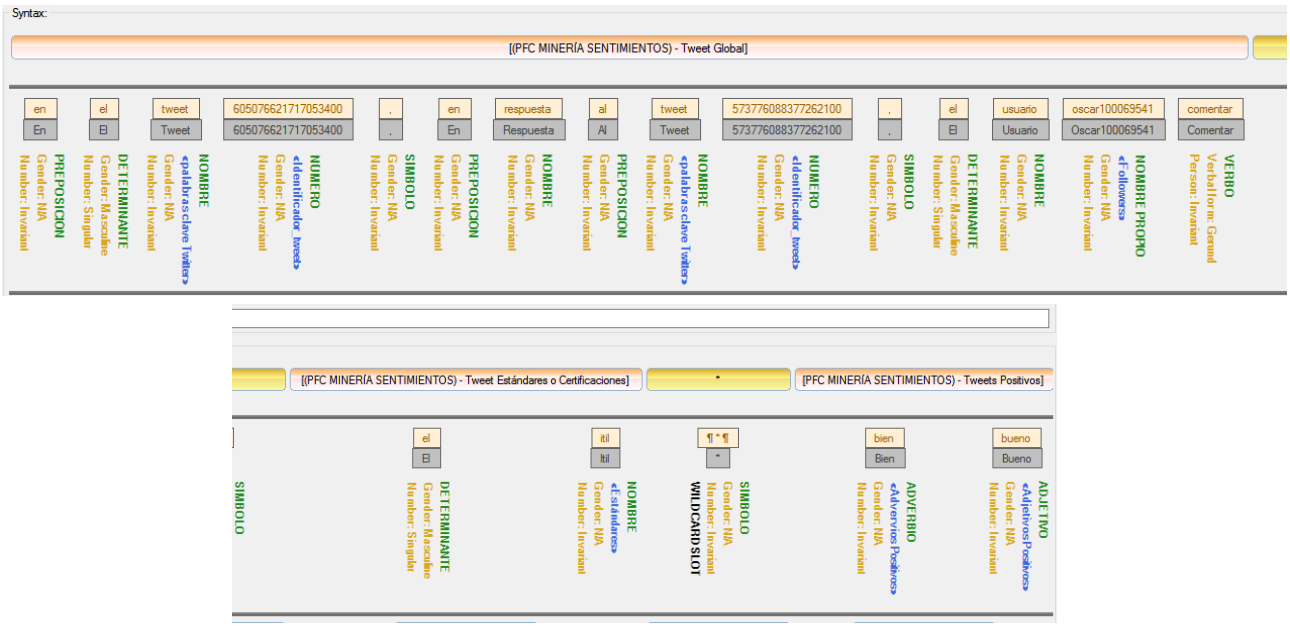

Ilustración 119 – Patrones – Patrón complejo estándar, positivo

El resultado de la ejecución para el *tweet*, es la aplicación del nuevo patrón. Patrón global positivo sobre estándares/certificaciones.

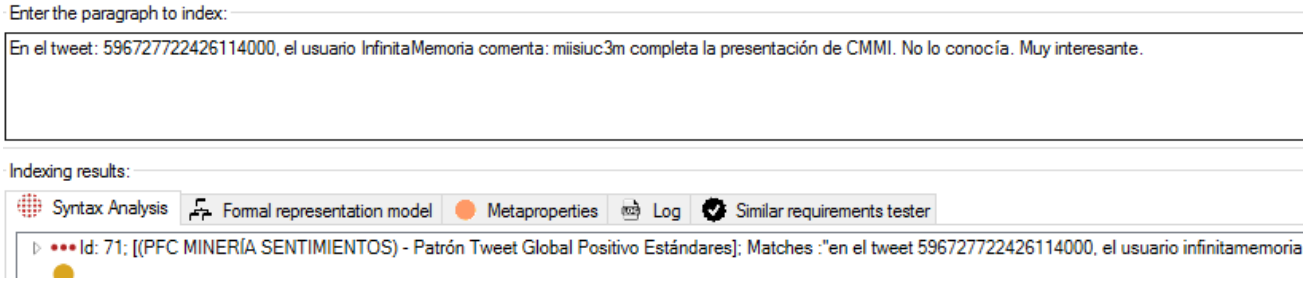

Ilustración 120 – Patrones – Aplicación patrón complejo positivo – estándar/certificación

6.5 Semántica

En la definición sintáctica de los *tweets* ya se ha realizado una aproximación a la semántica de los mismos.

El patrón global establece la regla básica de composición del *tweet* en lo relativo a sus elementos base:

- Identificador del *tweet*.
- Usuario que crea el *tweet*.
- *Si procede, tweet* que se responde.

El resto del *tweet* es realmente el texto escrito por el usuario. En este punto, se podría haber optado por una definición genérica de patrones que tuviese el único objetivo de recoger las posibilidades sintácticas de los *tweets.* Sin embargo, se ha dotado de algo más de inteligencia al proceso y se han definido patrones para

reconocer las estructuras sintácticas aportando el carácter del *tweet*. De ahí salen los patrones de *tweets* positivos, negativos, de opinión, etc.

Esta estructura de patrones facilita la creación de los elementos semánticos de la ontología (relaciones y meta-propiedades) ya que el trabajo de formalización dentro de knowledgeMANAGER se construye a partir de los patrones.

6.6 Relaciones

Las relaciones vendrán definidas por al menos dos elementos.

- Tipo de Relación / Verbo.

  El tipo de relación o un término a modo de verbo, define la semántica que se aplica en la relación.

- Elemento o elementos del patrón que se relacionan.

  Elemento o elementos que se relacionan. El elemento puede provenir de la propia sintaxis a través de los tokens que componen el patrón o pueden ser definidos como elementos fijos.

Gráficamente la herramienta los representa en forma de árbol, manteniendo el verbo como elemento principal y los elementos que componen la relación como nodos hoja.

6.6.1 Tipos de Relación

La herramienta viene precargada con un conjunto de relaciones, pero para la ejecución del proyecto se han creado otro conjunto que expresan la semántica que se quiere aplicar.

Se han creado las siguientes relaciones.

Relación	**Semántica**
Tweet creado por usuario	Representa la relación entre un *tweet* y el usuario que lo creó. El objetivo de esta relación es poder tener una relación de todos los usuarios que han creado *tweets*.
Tweet respondido en tweet por usuario	Representa la relación un *tweet*, el *tweet* al que responde y el usuario que publicó el *tweet.* Mediante esta relación se obtienen los *tweets* que responden a otros *tweets* así como el usuario que ha respondido.
Tweet estándar y certificación	Relación de los *tweets* que refieren a estándares o certificaciones.
Tweet negativo	Relación de los *tweets* con carácter negativo.
Tweet opinión	Relación de los *tweets* de opinión.
Tweet positivo	Relación de los *tweets* con carácter positivo.

Relación	Semántica
Tweet pregunta	Relación de los *tweets* que plantean una pregunta
Tweet sarcasmo	Relación de los *tweets* con carácter sarcástico.

Tabla 41 – Tipos de relaciones

En la herramienta las relaciones se crean dentro del modelo conceptual. Como el resto de elementos creados para el proyecto, se ha utilizado el prefijo (PFC MINERÍA SENTIMIENTOS) para diferenciarlos fácilmente de los pre-cargados en la herramienta.

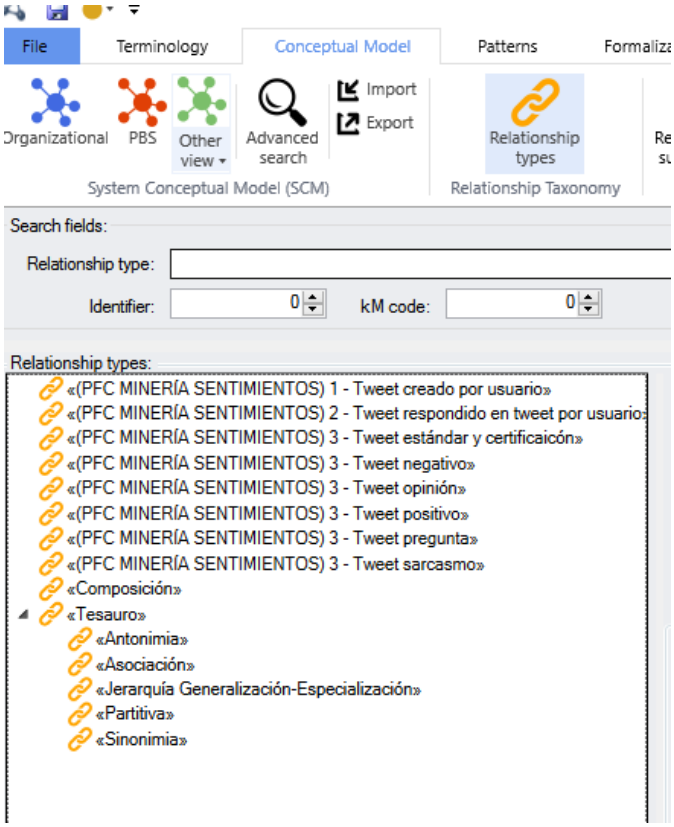

Ilustración 121 – Modelo conceptual – Tipos de relaciones

6.6.2 Relaciones

Como puede observarse, existe una estrecha relación entre los tipos de relaciones y el trabajo previo de creación de patrones. Mediante este mecanismo, la formalización de las relaciones se ha simplificado de forma que a partir de los patrones específicos de cada tipología de *tweet*, se aplican una o varias relaciones de forma clara.

Podemos diferenciar entre dos tipos de relaciones.

- Relaciones para obtener semánticas relativas a conceptos generales del *tweet*.

- *Tweet* creado por usuario.

- *Tweet* respondido en tweet por usuario.

Ambas relaciones se aplican sobre el patrón *Tweet* Global y debido a la estructura jerárquica de patrones, todos los patrones finales contienen a este sub-patrón. Esto provoca que cuando se indexen los *tweets*, las relaciones definidas a este nivel, se analicen para todos los *tweets*.

Veamos cómo se desarrolla el proceso de creación.

1. En primer lugar, se selecciona el patrón en el que se va a crear la relación en la pestaña de formalización.

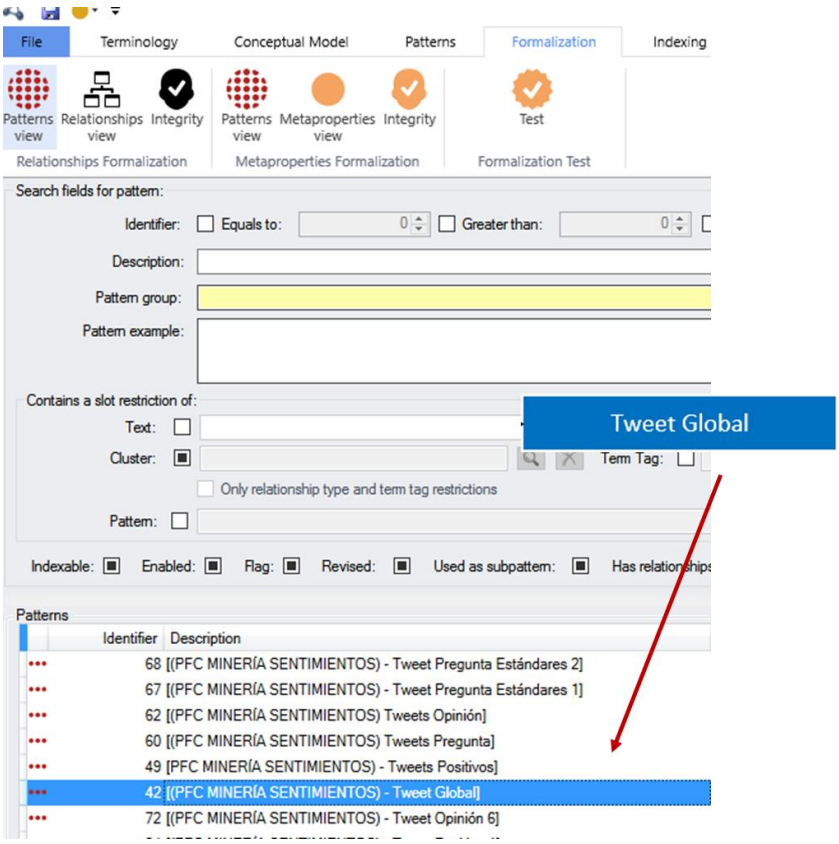

Ilustración 122 – Creación de relaciones sobre tweet global

2. Al entrar, el asistente nos muestra una primera regla vacía, en la que se pueden apreciar los elementos que la componen. El verbo o tipo de relación semántica y los términos sobre los que aplica la relación.

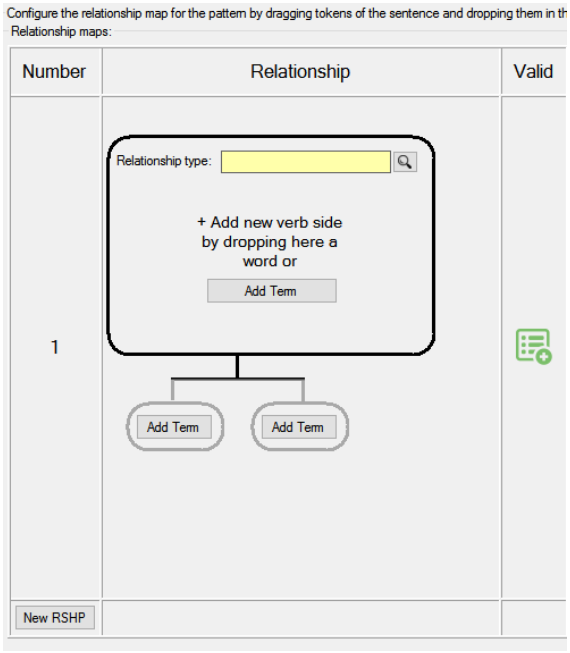

Ilustración 123 – Creación de relaciones – Relación en vacío

3. Seleccionamos el tipo de regla de las ya creadas (habrán de estar creadas previamente).
   En este punto también podría relacionarse un término de la ontología o un token del patrón. Aplicaría por lo tanto la semántica que el propio término tenga. Dado que en el proyecto pretendemos extraer una semántica contextualizada en Twitter, usuario que crea un *tweet* para el caso que estamos analizando, procedemos con la selección de la semántica creada.

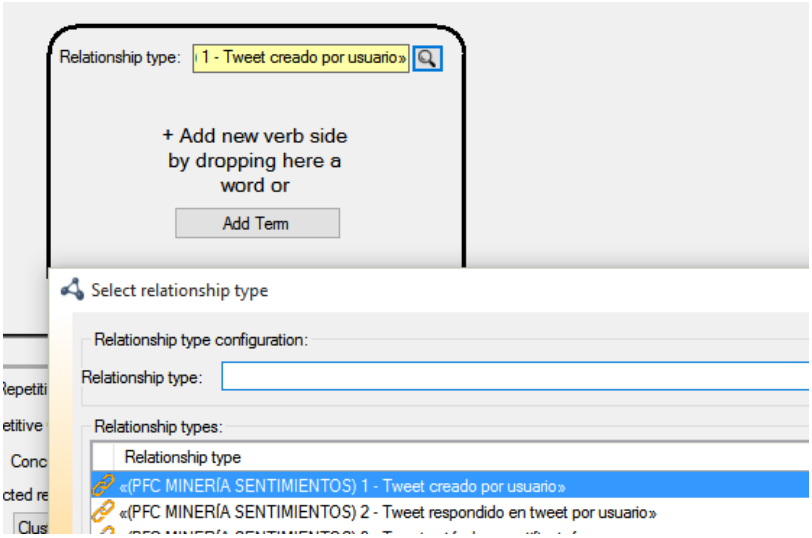

Ilustración 124 – Creación de relaciones – Selección del tipo de relación

4. Se seleccionan los términos que se relacionan. Para el caso que nos ocupa, el identificador del *tweet* y el usuario del *tweet*. Dado que esos ítems existen en el patrón sobre el que aplicamos la relación, se seleccionan directamente sobre el cluster en el que están agrupados.

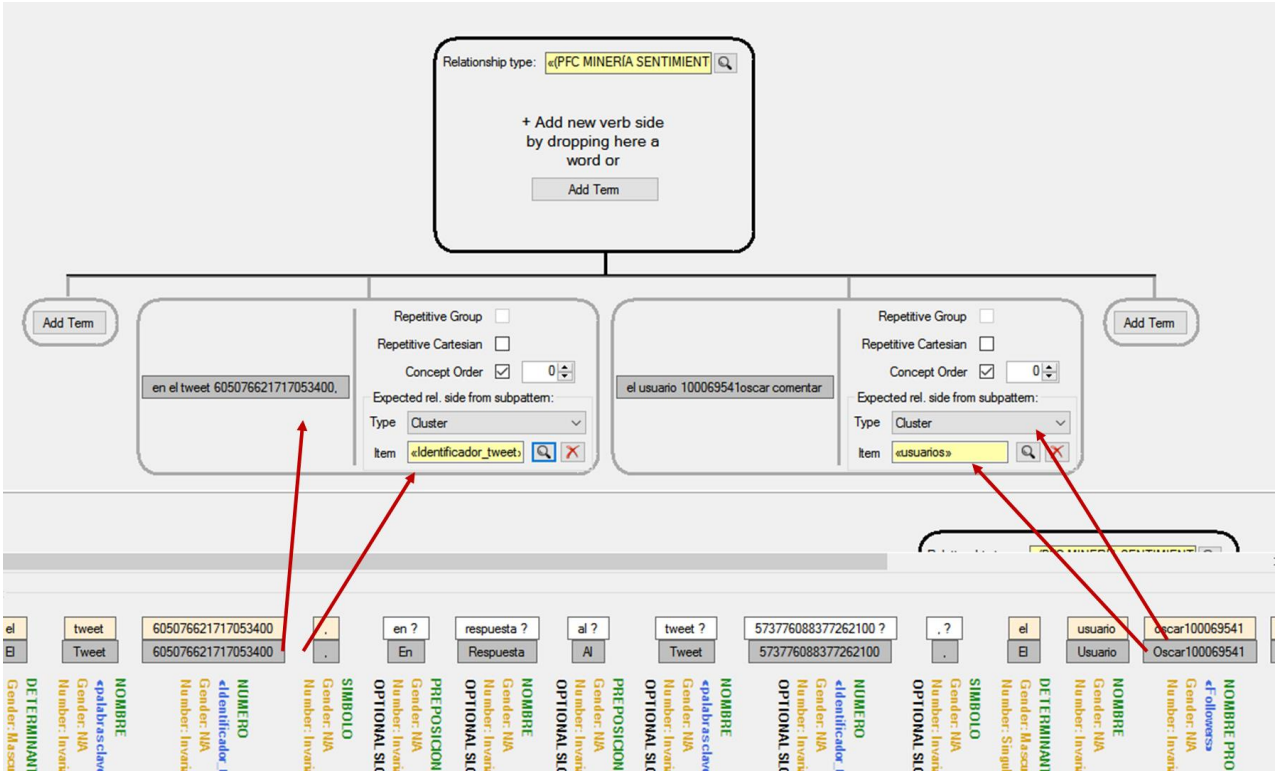

Ilustración 125 – Creación de relaciones – Patrón completo

Al utilizar los términos del patrón, cuando se indexe un *tweet* la relación será instanciada con los elementos términos particulares de ese *tweet*.

En el siguiente ejemplo puede verse como la aplicación de la relación instancia automáticamente el identificador del *tweet* y el usuario que lo generó.

Ilustración 126 – Creación de relaciones – Ejemplo aplicación relación

El ejemplo ha desarrollado la creación de la relación de *Tweet* creado por usuario sobre el patrón *Tweet* Global. La creación de la relación *Tweet* respondido en tweet por usuario se realiza de forma similar, hay que tener en cuenta que sobre un patrón pueden definirse múltiples relaciones.

Tal y como se explicó al inicio de este apartado, la relación debe tener al menos dos elementos, un verbo o tipo de relación y un término sobre el que aplica, pero no está limitada a un término, como se puede apreciar en este caso donde se llega hasta tres.

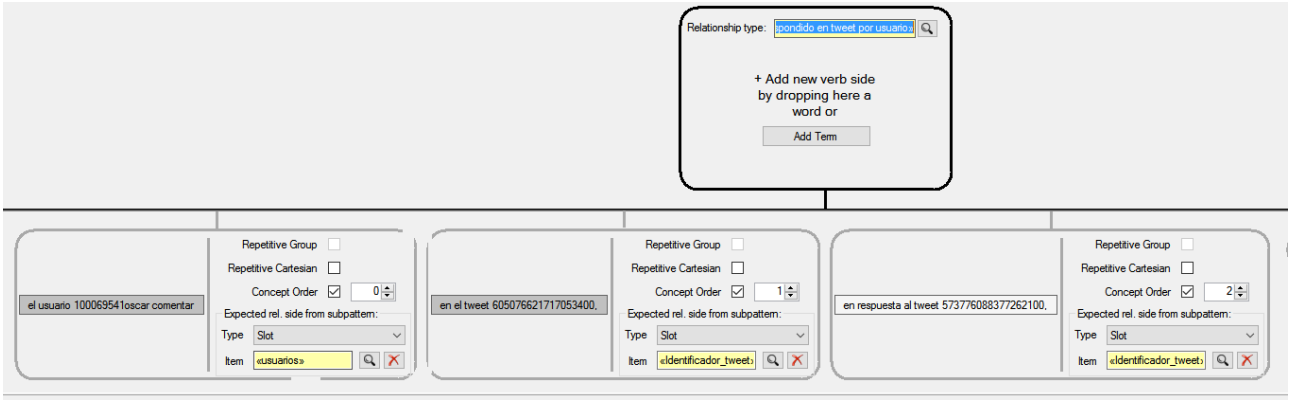

Ilustración 127 – Creación de relaciones – Patrón con tres términos

- Relaciones basadas en el carácter del *tweet*.

    - *Tweet* estándar y certificación
    - *Tweet* negativo
    - *Tweet* opinión
    - *Tweet* positivo
    - *Tweet* pregunta
    - *Tweet* sarcasmo

    La creación de todas estas relaciones es muy similar a la explicada con anterioridad pero cambiando el tipo de relación en cada caso.

    En todos los casos la relación indexa el identificador del *tweet* y el usuario que lo ha publicado. En la relación *tweet* estándar y certificación se muestra además el tipo de estándar o certificación sobre el que trata el *tweet*.

    Veámoslo con un ejemplo de indexación.

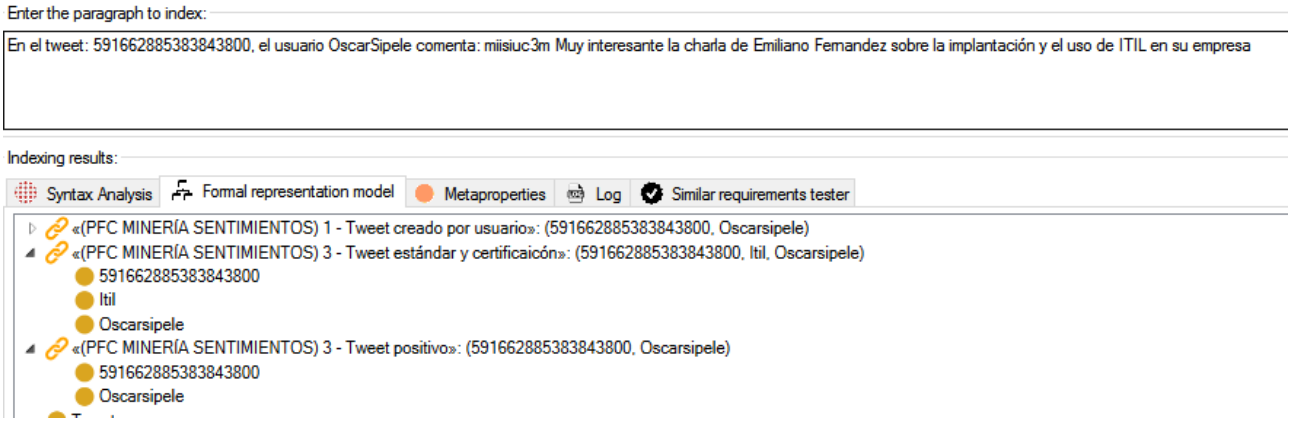

Ilustración 128 – Creación de relaciones basadas en carácter del tweet

La dificultad en la creación de estas relaciones estriba en la elección del patrón sobre el que aplicarlas.

Al igual que con las relaciones basadas en datos generales del *tweet*, el modelo de patrones implementado nos proporciona una guía de cara a la implementación de relaciones.
En este caso, la lógica nos hace pensar que los patrones sobre los que debe aplicarse la relación, son los denominados patrones finales descritos en el punto 6.4.3 Patrones Finales.

- Patrón *Tweet* Global Positivo
  - ⇨ Relación *Tweet* Positivo
- Patrón *Tweet* Global Negativo
  - ⇨ Relación *Tweet* Negativo
- Patrón *Tweet* Global Sarcasmo
  - ⇨ Relación *Tweet* Sarcasmo
- Patrón *Tweet* Global Pregunta.
  - ⇨ Relación *Tweet* Pregunta
- Patrón *Tweet* Global Estándares y Certificaciones.
  - ⇨ Relación *Tweet* Estándar y Certificación
- Patrón *Tweet* Global Sarcasmo.
  - ⇨ Relación *Tweet* Sarcasmo

De esta forma, dado que todos los *tweets* aplican sobre uno de estos patrones, todos los *tweets* quedarán clasificados en base a su carácter.

¿Y qué pasa con los patrones complejos?

Recordemos que se han definido patrones complejos para la indexación de aquellos *tweets* que tienen dos características, por ejemplo un *tweet* que es positivo sobre un estándar.
La respuesta a la cuestión es sencilla, en los patrones complejos, implementamos dos relaciones, una por cada sub-patrón que incluye.

Este fue el enfoque a la hora de implementar estos patrones pero nos encontramos con un bug en knowledgeMANAGER, bug que ya está resuelto en la versión superior y que paso a describir.

El problema surge cuando se intenta seleccionar un término del patrón y el patrón está compuesto por dos sub-patrones unidos con la cláusula *or*.
La herramienta sólo reconoce uno de los dos patrones y cuando se indexa un elemento, si el token que tiene que indexar se encuentra en el segundo patrón, no se indexa correctamente.

Veámoslo con un ejemplo, patrón de *tweets* positivos y que hacen referencia a estándares.

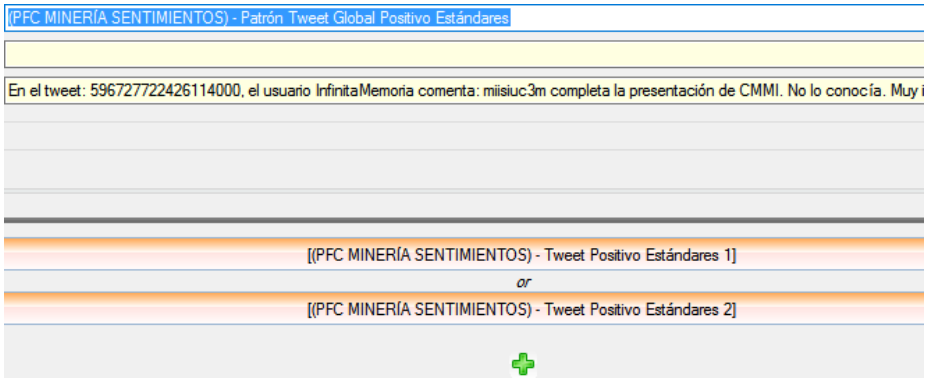

Ilustración 129 – Patrón tweet positivo estándar 1 y 2

Este patrón se compone de dos sub-patrones con el objetivo de indexar correctamente los *tweets* independientemente de si en primer lugar se expresa el carácter positivo y después se menciona el estándar/certificación, o se hace al revés.

> *"En el tweet: 596727722426114000, el usuario InfinitaMemoria comenta: miisiuc3m completa la presentación de CMMI. No lo conocía. Muy interesante."*
>
> *"En el tweet: 596727722426114000, el usuario InfinitaMemoria comenta: miisiuc3m Muy interesante y completa la presentación de CMMI. No lo conocía."*

Cuando estamos creando la relación, a la hora de indicar el token del patrón que va a indexarse, la herramienta no tiene en cuenta el segundo patrón.

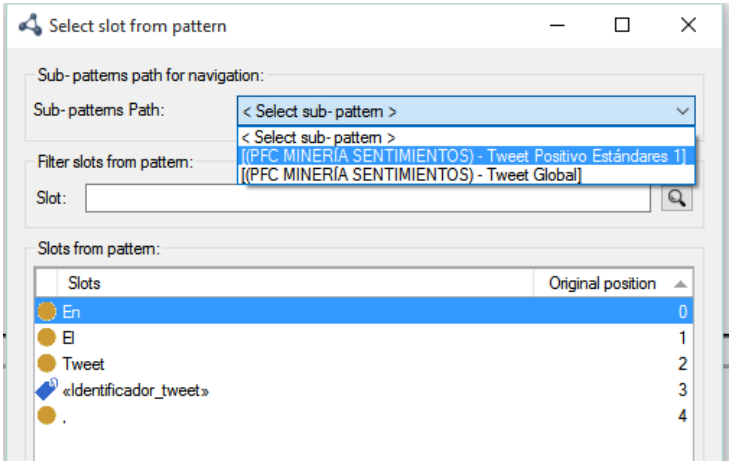

Ilustración 130 – Error al seleccionar patrón

Esto provoca que al realizar la indexación, sólo en uno de los *tweets* mostrados, la relación semántica se aplique.

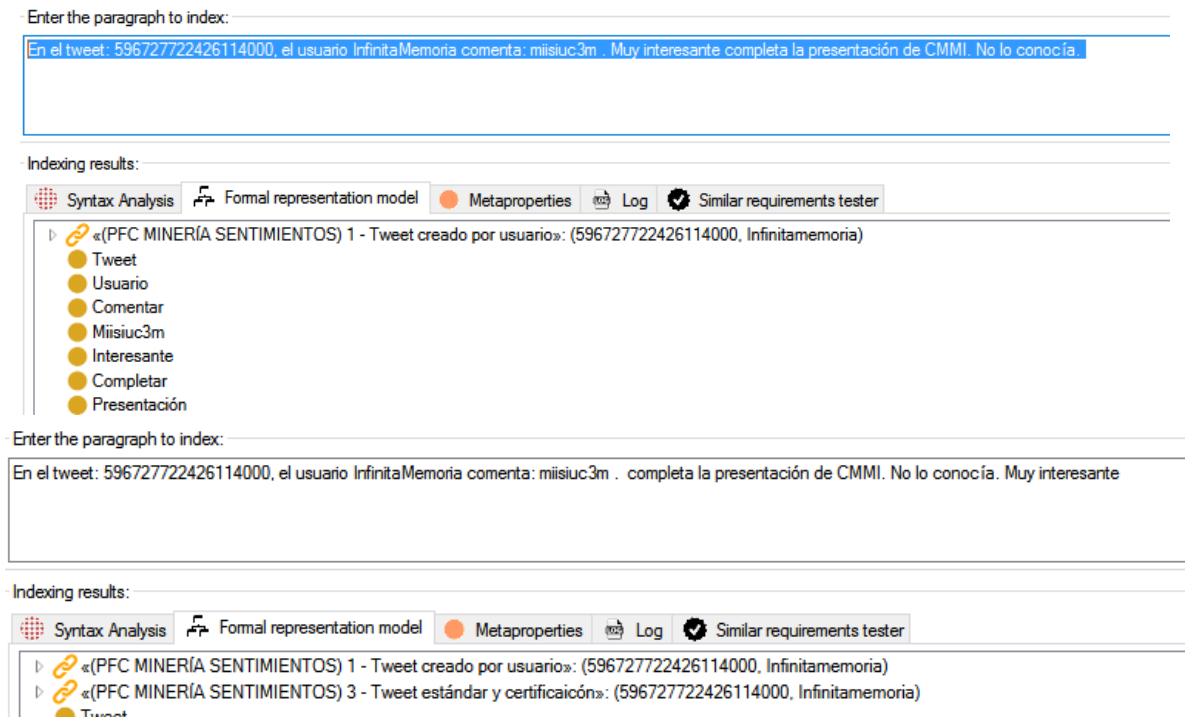

Ilustración 131 – Error al indexar

Para evitar este problema, la solución aplicada ha consistido en la creación de las relaciones en los patrones inferiores.

En el ejemplo anterior, en lugar de crearse las relaciones en el patrón *Tweet* Global Positivo Estándares, se ha creado en cada uno de sus sub-patrones: *Tweet* Positivo Estándares 1 y *Tweet* Positivo Estándares 2.

El siguiente diagrama muestra el mapa de patrones y relaciones creadas en cada patrón.

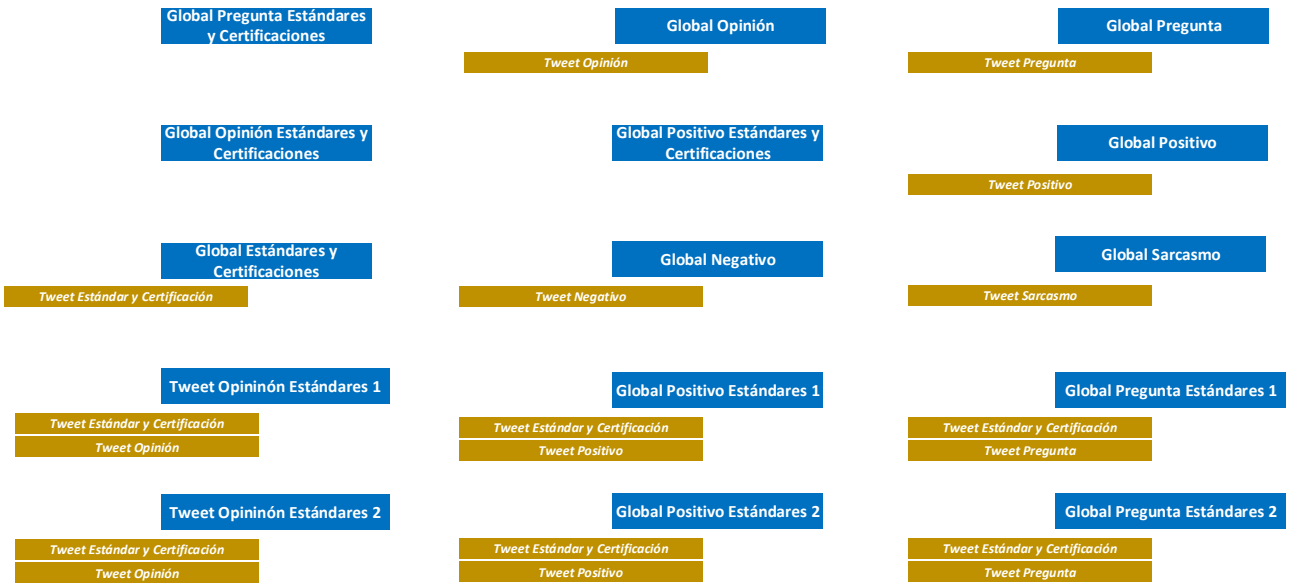

Ilustración 132 – Mapa de relaciones según el carácter del tweet

6.7 Meta-propiedades

Continuando con el aporte de semántica a los *tweets*, el siguiente paso es la definición de atributos y cualidades de los *tweets*. La forma de gestionarlo en knowledgeMANAGER es mediante la definición de meta-propiedades asociadas a los patrones. Propiedades que en el proceso de indexación del *tweet*, se instancian con los valores que corresponda.

Al igual que con las relaciones, la estructura de patrones definida, facilita la creación de meta-propiedades.

De forma similar al enfoque de las relaciones, se han definido dos tipos de meta-propiedades.

- Propiedades relativas a conceptos generales del *tweet.*
    - ID del *tweet*.

      Recogerá el identificador del *tweet*.
    - Usuario del *tweet*

      Usuario que ha publicado el *tweet*.
    - Es respuesta.

      Indicará si el *tweet* responde a otro *tweet*.
    - ID del *tweet* respondido.

      Para los *tweets* de respuesta, recogerá el identificador del *tweet* respondido.

  Dada la estructura jerárquica de patrones creada, asociar la meta-propiedad a un patrón sencillo hace que la hereden los patrones compuestos que dependen de él. Por lo tanto cada una de estas propiedades ha sido asociada al patrón sencillo que corresponde.

  Se detalla a continuación el proceso de creación.

  1- Partimos de la pestaña de formalización, de la opción de meta-propiedades.

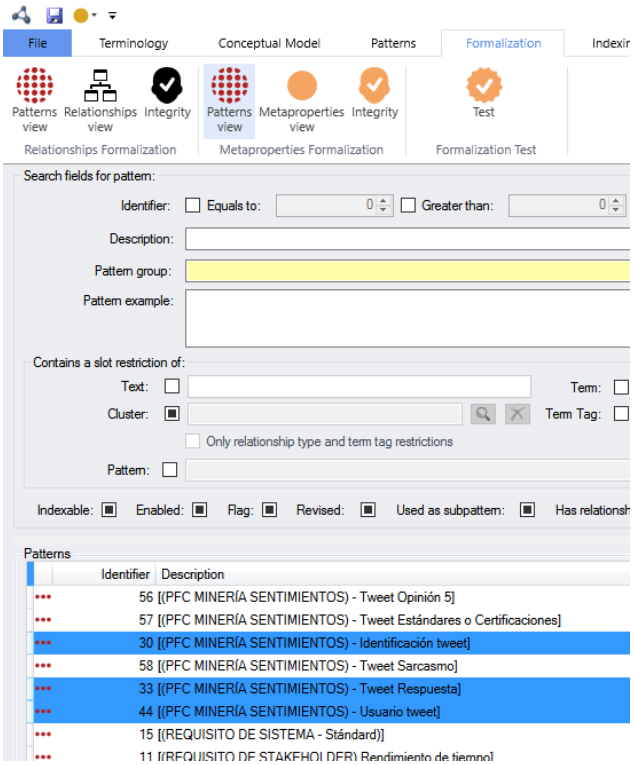

Ilustración 133 – Creación de meta-propiedades base

2- Seleccionando el patrón sobre el que se quieren crear las propiedades, se muestra un formulario para añadir tantas propiedades como se necesiten.

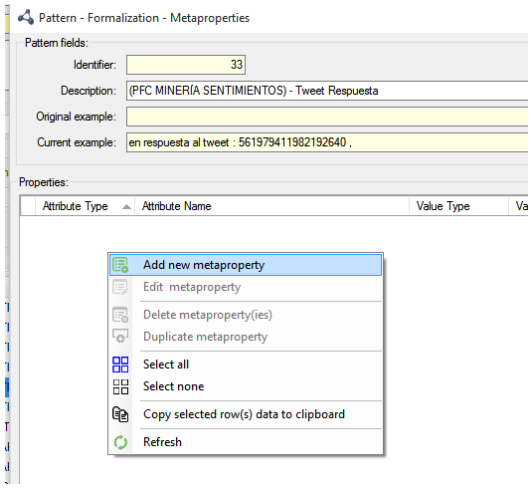

Ilustración 134 – Formulario meta-propiedades

3- La meta-propiedad pueden estar definida por valores fijos o instanciarse con el contenido de uno o varios slots del patrón.

En la ilustración puede observarse la creación de las dos meta-propiedades asociadas al patrón *Tweet* Respuesta.

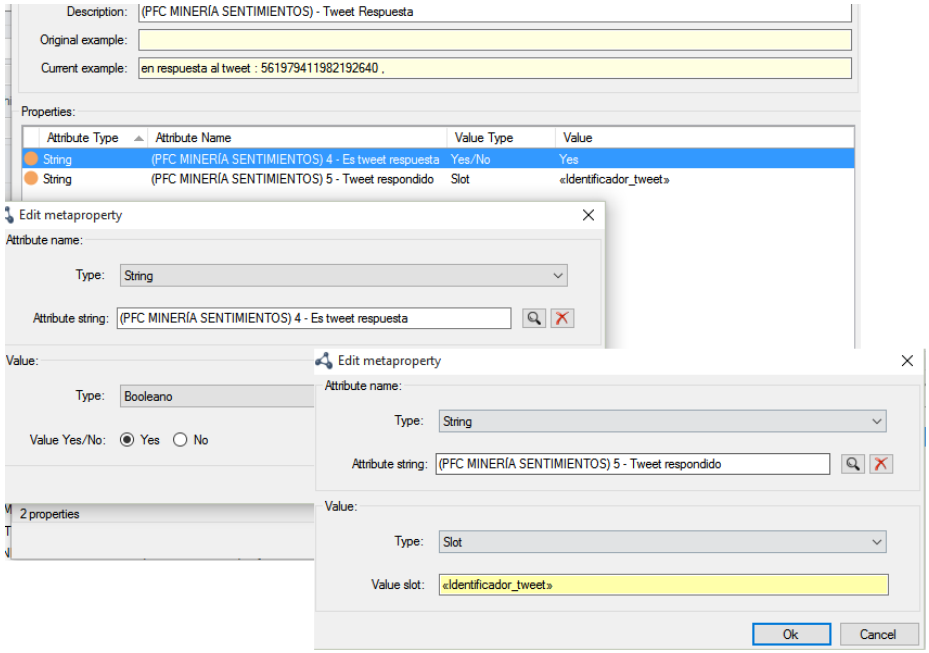

Ilustración 135 – Meta-propiedades de respuesta

Todos los *tweets* que contengan este patrón, tendrán instanciada la propiedad "Es *tweet* respuesta" a "Yes" y además tendrán el identificador del tweet al que se responde en la propiedad "*Tweet* respondido".

Podemos apreciar claramente el resultado de la indexación de un *tweet* respuesta en la siguiente imagen.

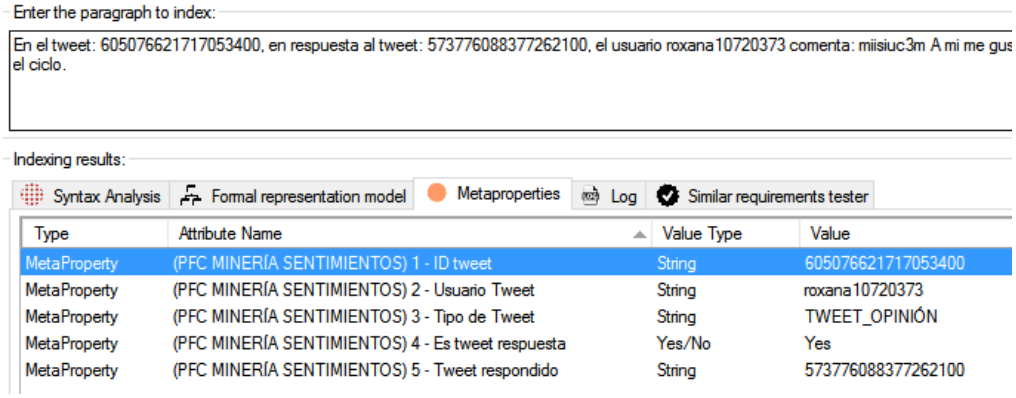

Ilustración 136 – Meta-propiedades de respuesta

- Propiedades relativas al carácter del *tweet*.
  Con el objetivo de calificar el *tweet* y detectar en la indexación el carácter del *tweet*, se crea una meta-propiedad más, "Tipo de *tweet*".

La propiedad tomará un valor u otro en función del carácter del *tweet*.
Navegando por cada uno de los patrones finales y complejos (patrones hoja en definitiva), se va instanciando la propiedad.

Como puede observarse, los *tweets* complejos, tienen dos propiedades asociadas, una por cada uno de los sentimientos que expresan.

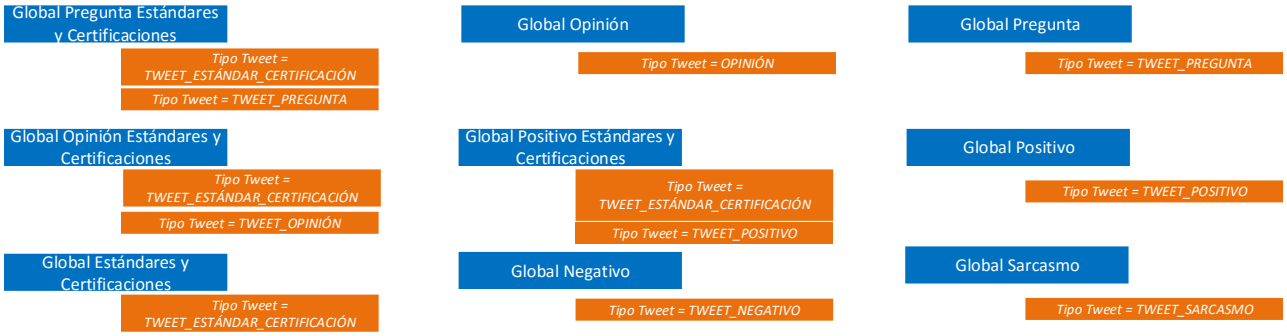

Ilustración 137 – Meta-propiedades de respuesta

6.8 Indexación

Concluimos con esta tarea el proceso de minería de sentimientos apoyado en la ontología implementada.

Se inició el proceso completando los términos de la lengua castellana disponibles en la gramática de lenguaje natural precargado de la herramienta incorporando aquellos relativos a Twitter.

Se continuó con la agrupación de los términos del estudio en clústeres, clústeres que serían utilizados en siguiente paso, los patrones que definen la sintaxis de los *tweets* recogidos y transformados.

El proceso de creación de los patrones ha sido de especial relevancia, ya que además de definir la sintaxis, se han creado con una orientación algo semántica, de forma que la fase posterior de dotación de semántica a la ontología mediante la creación de relaciones entre patrones y definición de meta-propiedades, se ha apoyado en la estructura de patrones definidos.

Para concluir con el trabajo de minería, se procede con la indexación de los *tweets*, esto es, análisis e interpretación de los *tweets* aplicando las relaciones y extrayendo las meta-propiedades que le aplican.

En la fase de transformación y preparación, mediante el desarrollo y utilización de *transformTwitterData* se procedió a transformar todos los *tweets* extraídos mediante el API de Twitter a lenguaje natural. El resultado de esa extracción es quien va a nutrir el proceso de indexación.

Durante la explicación de la implementación de los patrones, las relaciones y las meta-propiedades ya se ha mostrado algún ejemplo de indexación para ejemplificar el elemento que se estuviese desarrollando en ese momento. Entraremos a continuación en el análisis en profundidad del proceso y las posibilidades de indexación.

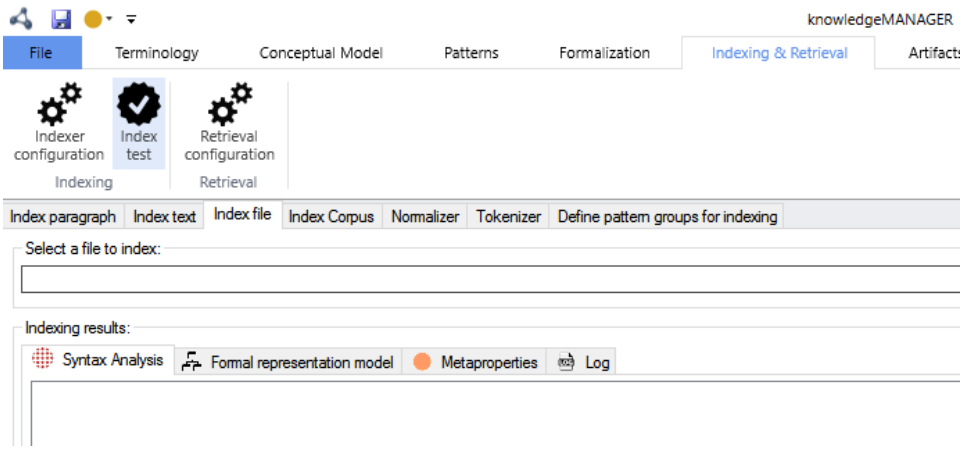

Ilustración 138 – Proceso de indexación

Aunque la herramienta proporciona la posibilidad de analizar un *tweet* o varios en concreto (opciones Index paragraph y Index text), para el estudio que nos ocupa, hemos elegido la opción Index file. Se le pasa un fichero con la información a indexar y ofrece los siguientes resultados:

- Análisis sintáctico.
  Se muestran los patrones que aplican a los elementos analizados.

  Por ejemplo:

  “
  En el tweet: 592643238726598700, el usuario 100069541Oscar comenta: miisiuc3m Muy interesante la ponencia de Itil por parte de Emiliano!
  “

  Estructura de patrones que aplica sobre el *tweet*.

  Tweet Global Positivo Estándar
  Tweet Positivo Estándares 2
  Tweet Global
  Identificación tweet
  Usuario tweet
  Tweet Positivo
  Tweet Positivo 1
  Tweet Estándares o Certificaciones.

  Como puede apreciarse, se muestra cada patrón aplicado sobre cada *tweet* analizado. También podemos ver cómo se forma cada estructura en base a sub-patrones.

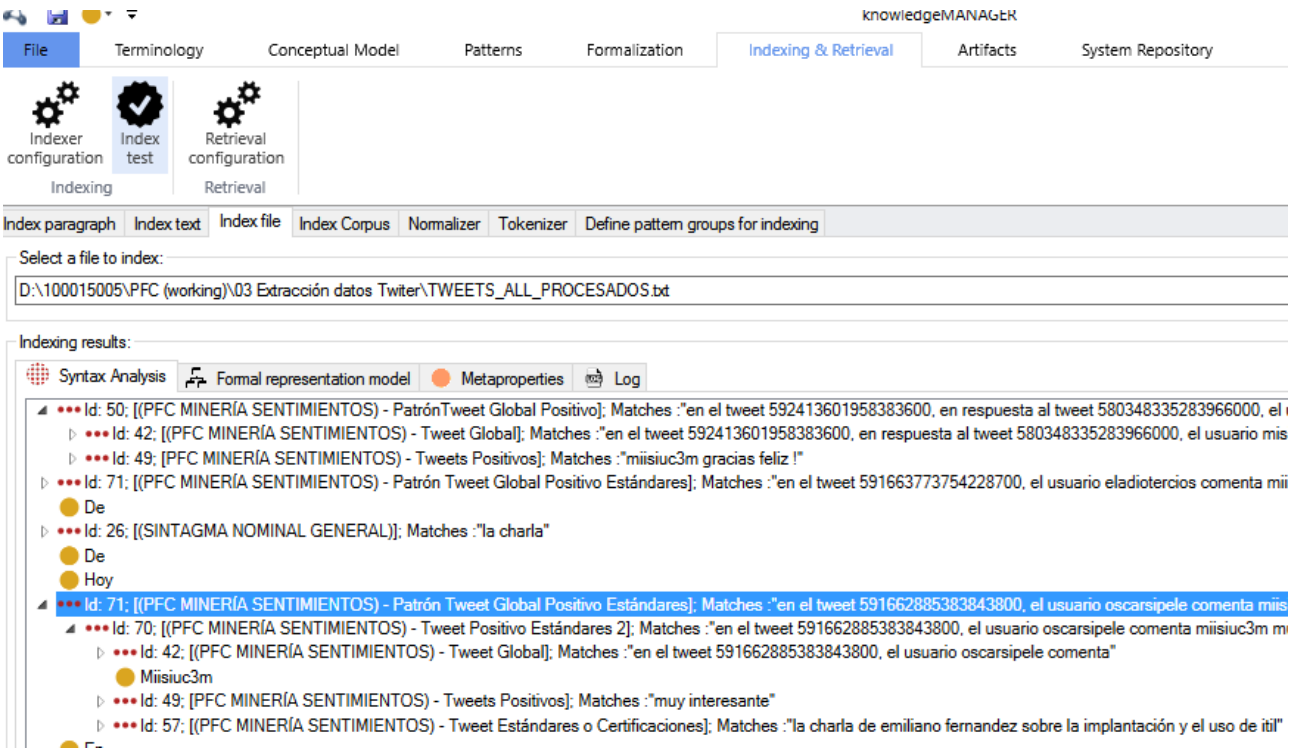

Ilustración 139 – Indexación - Sintaxis

- Representación formal.
  Se muestran todas las relaciones que se han encontrado instanciando los valores de las mismas.

  Esta vista resulta muy interesante ya que de ella se extrae conocimiento en base a las relaciones definidas. Podemos saber por ejemplo qué *tweet* ha creado cada usuario, en qué *tweet* se responde un determinado *tweet* (y qué usuario lo responde), qué *tweets* y usuarios han opinado sobre estándares o certificaciones, sobre qué estándar o certificación, qué *tweets* son de opinión y qué usuarios lo han publicado, positivos, negativos, etc.

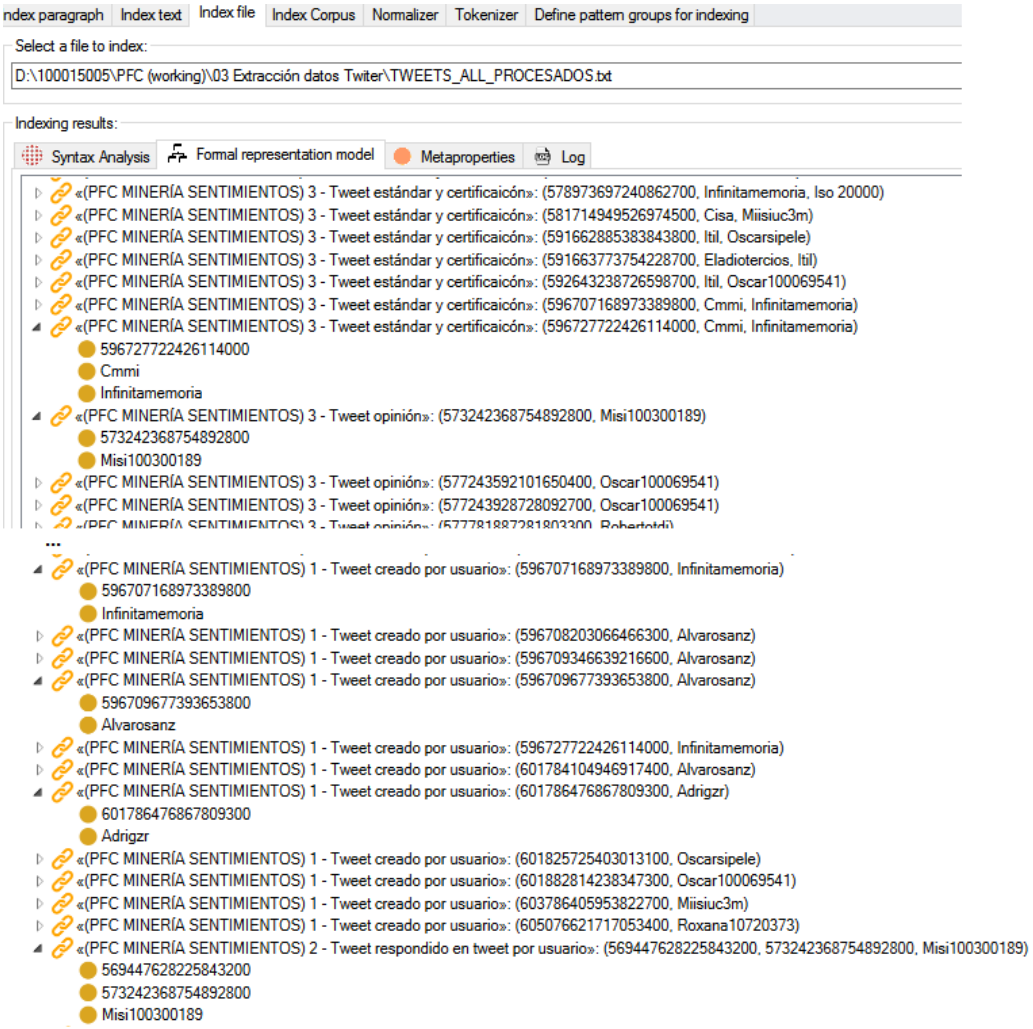

Ilustración 140 – Indexación – Semántica - Relaciones

- Meta-propiedades.
  Toda meta-propiedad extraída del análisis, instanciadas con el valor correspondiente.

  Nuevamente se trata de un proceso de dotación de semántica al *tweet*. Al ser analizados conjuntamente, podemos extraer todos los identificadores de *tweet* así como los usuarios que los publican, si son *tweets* de respuesta (sobre qué otro *tweet*) y finalmente el tipo de *tweet*. Esta última información es muy relevante a la hora de analizar los sentimientos expresados por los alumnos.

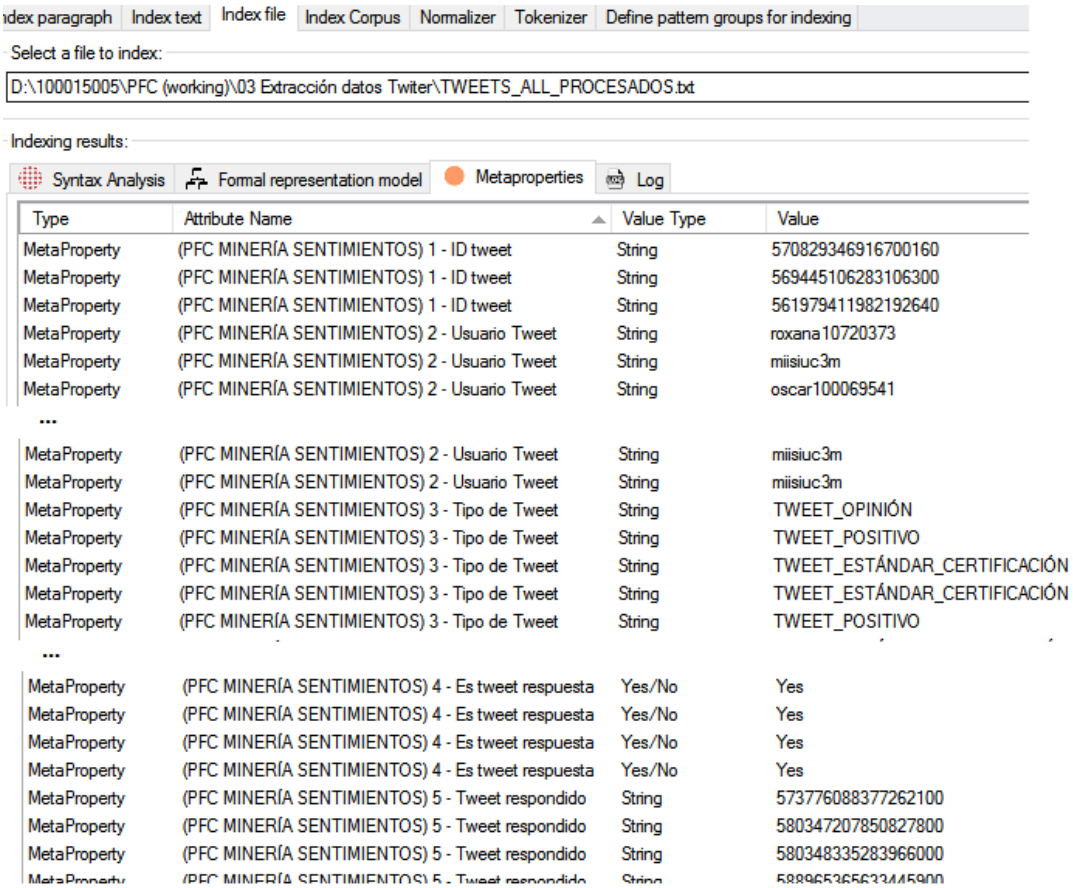

Ilustración 141 – Indexación – Semántica – Meta-propiedades

Capítulo 7

Resultados

Se muestra un análisis de algunos indicadores en base al resultado del proceso final de indexación con el que se ha concluido el proceso de minería.

7. Resultados

Iniciamos el proceso de minería mediante la extracción de datos a través del API REST de Twitter, continuamos el proceso mediante la transformación y preparación de los datos. La siguiente fase del proceso de KDD ha sido la propia minería de datos.

Todo ello sin olvidar el objetivo final del proyecto, convertir las expresiones de los usuarios y docentes en conocimiento.

Con ese objetivo, en este capítulo desarrolla un estudio estadístico de los resultados de la indexación.

Ilustración 142 – Proceso de KDD

7.1 Resultados del Análisis

A través de la opción *Artifacts* de knowledgeMANAGER analizamos el resultado del análisis tras la indexación de todos los *tweets.*

La valoración estadística nos muestra que en total se han analizado 66 *tweets*. Hay que recordar en este punto que los *tweets* relevantes para el estudio han sido las menciones y los *tweets* publicados por miisi_uc3m tal y como ya se describió en el apartado 5.1.1 Recolección de Datos

De los 66 *tweets*, 30 son respuesta sobre otros *tweets*, un 45%.

Desde el punto de vista del carácter de los *tweets*, un 27% mencionan estándares o certificaciones y otro 27% son de carácter positivo, representando entre las dos tipologías un 54% del volumen total.

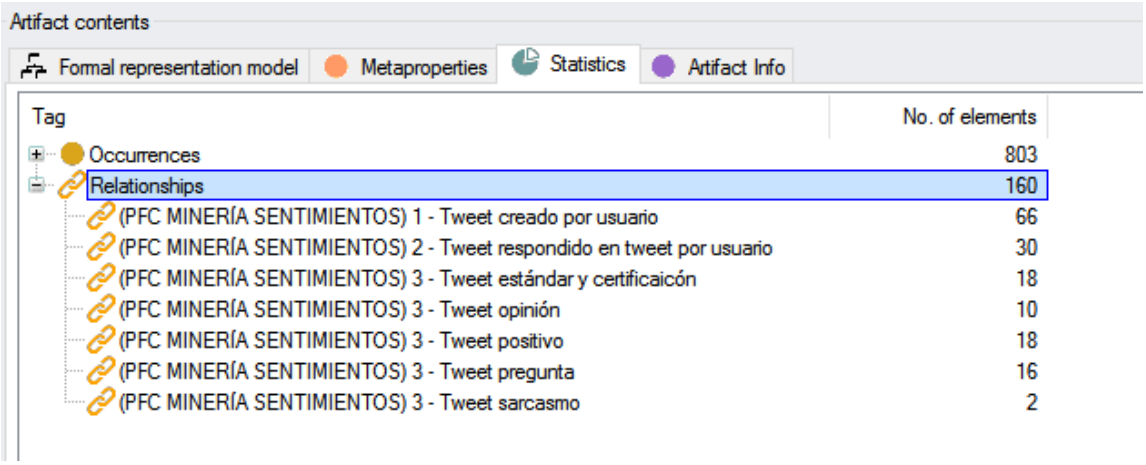

Ilustración 143 – Estadísticas de las relaciones

Por el contrario, la tipología menos representada de *tweets* son los sarcásticos con sólo 2 ocurrencias y negativos que no se han considerado ninguno. Y digo considerado porque sí existe un *tweet* que se ajusta a un patrón negativo.

"En el tweet: 581561754993487900, el usuario eladiotercios comenta: miisiuc3m Buen resumen de la asignatura de auditoría por robertotdi. Lástima que se haya visto afectada por los problemas técnicos".

Como puede observarse, el *tweet* contiene una expresión negativa "Lastima que se haya visto afectada por problemas técnicos" pero dado que el *tweet* realmente representa una valoración positiva, en la indexación lo he considerado como positivo.

Estos casos en los que existe cierta ambigüedad, se han resuelto mediante el uso de peso en los patrones.

Para aquellos casos en los que la expresión se ajusta a dos representaciones sintácticas, el mecanismo que discrimina entre una opción u otra, es el peso que tenga la regla sintáctica, el patrón.

Si vemos la indexación del *tweet*, realmente vemos que se reconocen dentro del texto la expresión positiva: "buen resumen" y la expresión negativa "lástima qué".

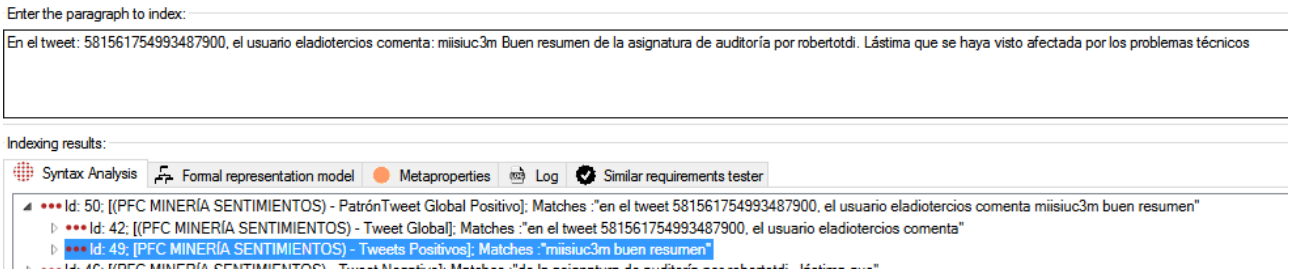

Ilustración 144 – Indexación tweet ambiguo

Pero dado que los patrones positivos tienen un peso mayor que los negativos, el *tweet* ha sido clasificado finalmente como positivo, ajustándose al patrón *tweet* global positivo.

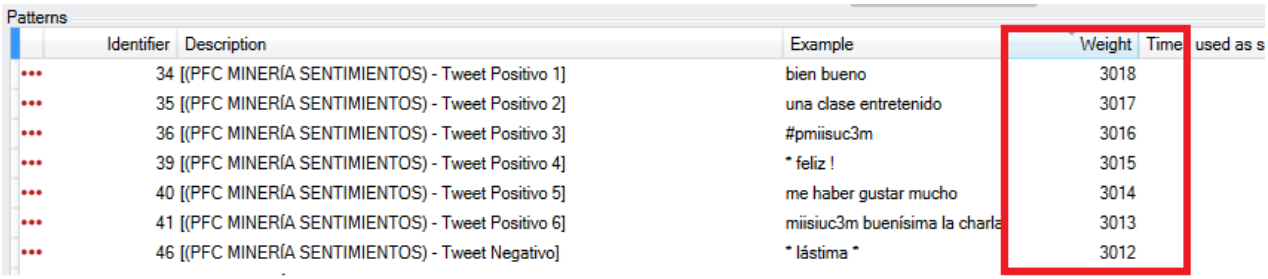

Ilustración 145 – Asignación de pesos a patrones

Se muestran a continuación otros resultados estadísticos en base al análisis de las relaciones y meta-propiedades.

7.1.1 Tweets por usuario

Mediante la relación "*Tweet* creado por usuario", obtenemos el número de *tweets* que ha realizado cada usuario.

El usuario con mayor actividad, ha sido miisiuc3m, el usuario gestor del perfil. Su actividad representa el 36% del total

Como cabía esperar y como patrocinador del perfil, su actividad es de todo tipo.

miisiuc3m	**24**
NO RESPUESTA	**11**
ESTÁNDAR Y PREGUNTA	5
NEUTRO	4
POSITIVO	1
PREGUNTA	1
RESPUESTA	**13**
NEUTRO	1
OPINIÓN	4
POSITIVO	2
PREGUNTA	6

Tabla 42 – Tweets miisiuc3m

En la siguiente gráfica se muestra la distribución de *tweets* por usuario.

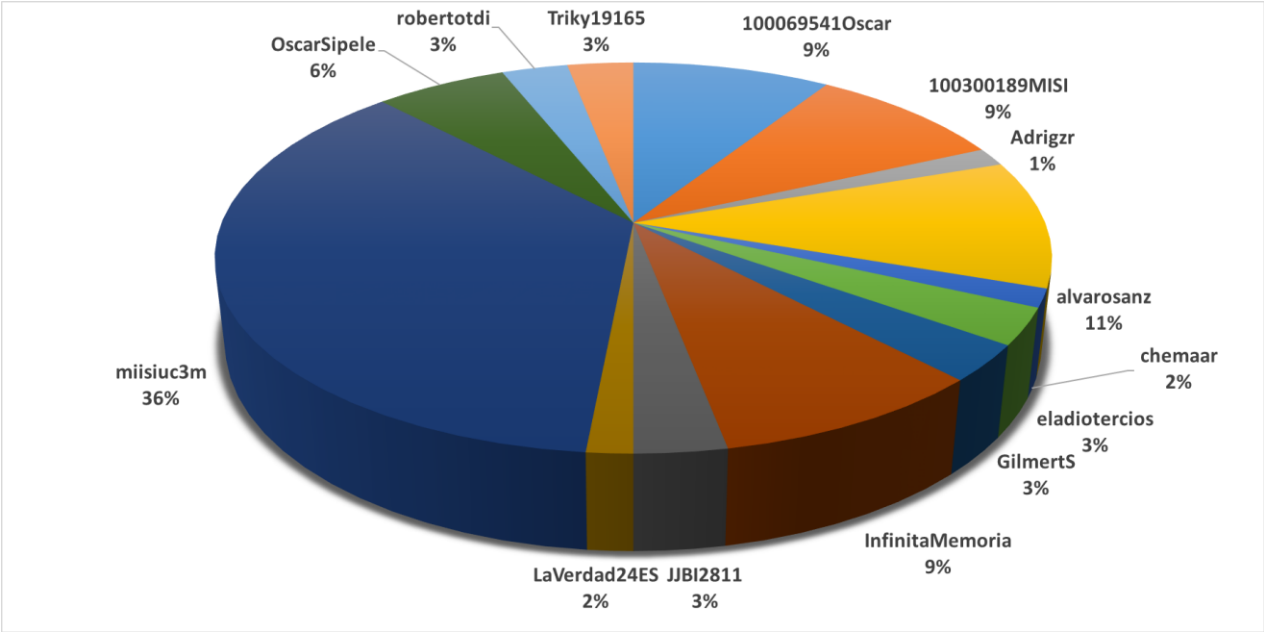

Ilustración 146 – Tweets por usuario

Los siguientes usuarios en volumen de actividad han sido 100300189MISI e InfinitaMemoria. Ambos con seis *tweets*.

100300189MISI	**6**
RESPUESTA	**6**
ESTÁNDAR	2
NEUTRO	1
OPINIÓN	2
POSITIVO	1
InfinitaMemoria	**6**
NO RESPUESTA	**6**
ESTÁNDAR	3
ESTÁNDAR Y POSITIVO	1
NEUTRO	1
POSITIVO	1

Tabla 43 – Tweets 100300189MISI e InfinitaMemoria

7.1.2 Tweets respuesta

Un dato interesante de medir es el número de *tweets* que son respuesta de otros *tweets* en tanto en cuanto puede relacionarse con el interés que genera cada publicación.

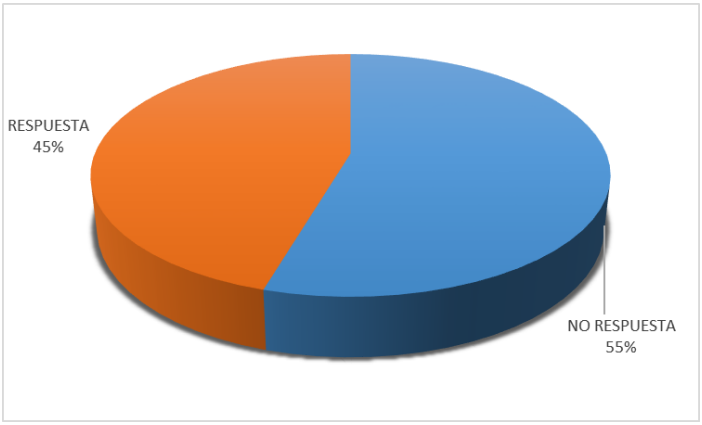

Ilustración 147 – Tweets respuesta

Los *tweets* que han tenido más respuestas, con tres cada uno, son los siguientes. Todos planteados por miisiuc3m.

“En el tweet: 573776088377262100, el usuario miisiuc3m comenta: Cual es el tema que mas os gusta de ITIL?”.

“En el tweet: 577670597263622100, el usuario miisiuc3m comenta: Qué pensáis de los planes de contingencia vs continuidad? Cual es vuestra interpretación de ITIL? Cual es vuestra opinión personal?”.

“En el tweet: 573776179808829440, el usuario miisiuc3m comenta: Se aplica ITIL en más empresas que habéis trabajado?”.

7.1.3 Tipos de tweets

En la introducción de las conclusiones ya pudimos ver el reparto del tipo de *tweets*.

Sobre dicho reparto, se han incluido algunos elementos más de análisis.

- *Tweets* que no se ajustan a ninguna clasificación e tipo, se han llamado *tweets* neutros.

- *Tweets* que se ajustan a dos clasificaciones. En el primer análisis se mostraban *tweets* de cada tipo en bruto, pero podemos afinar el análisis mostrando qué *tweets* se ajustan a dos clasificaciones, p.e. *tweets* positivos y relativos a estándar o certificación. Son los *tweets* que aplican a patrones complejos.

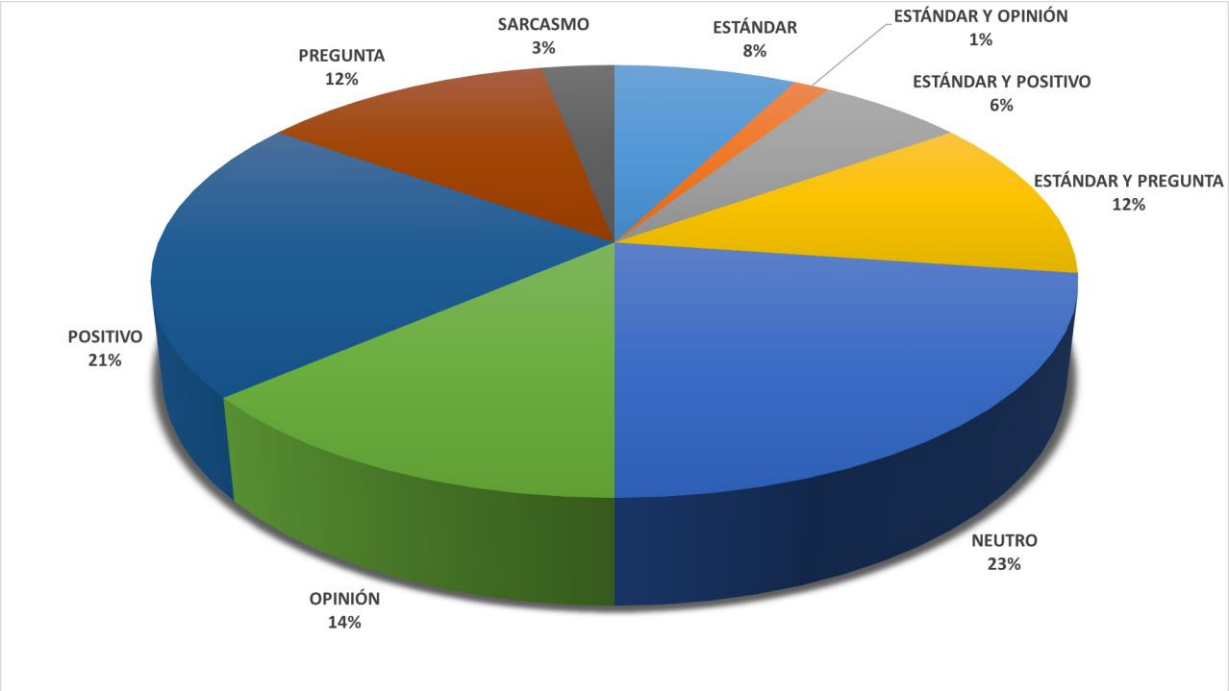

Ilustración 148 – Tipos de tweet

Una vez incluida esta clasificación, vemos que el 23% de los tweets son considerados como neutros.

Aquí mayormente encontramos tweets con comentarios generales:

“En el tweet: 578988439347003400, el usuario InfinitaMemoria comenta: miisiuc3m tal y como son las clases en el aula multimedia, las presentaciones y las revisiones tienen que estar más ceñidas al tiempo“.

“En el tweet: 580348335283966000, en respuesta al tweet: 573241930320113660, el usuario miisiuc3m comenta: 100300189MISI miisiuc3m tendréis la información en Aula Global”.

O tweets respuesta a otros tweets que finalizan una conversación.

“En el tweet: 574159981102120960, en respuesta al tweet: 573776179808829440, el usuario Triky19165 comenta: miisiuc3m Mi experiencia es en PYMES y no he tenido la oportunidad de utilizar todavía el framework a nivel laboral.”.

Que es una respuesta al tweet.

“En el tweet: 573776179808829440, el usuario miisiuc3m comenta: Se aplica ITIL en más empresas que habéis trabajado?”.

También se incluye en esta clasificación *tweets* cuyo contenido no puede ser analizado en lo relativo a su tipología.

En este ejemplo podemos ver un *tweet* que posiblemente fue publicado por error, ya que no incluye contenido.

“En el tweet: 588968537726836700, en respuesta al tweet: 588965365633445900, el usuario JJBI2811 comenta: JJBI2811 miisiuc3m”

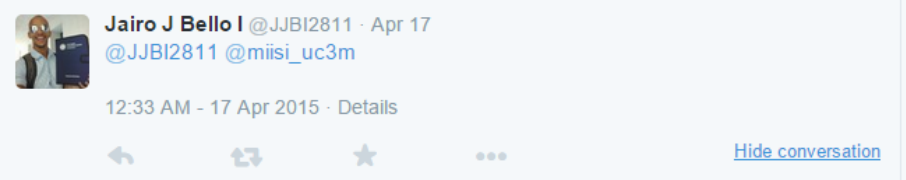

Ilustración 149 – tweet sin contenido

O que no aplican sobre ningún elemento de la asignatura. Estos *tweets* podrían haber sido eliminados del estudio en la fase de recolección de datos, sin embargo he preferido mantenerlos para ver el resultado de su indexación. Como debe ocurrir, no han sido clasificados por ningún patrón de tipo.

“En el tweet: 588447878005194800, el usuario xisalufufufa comenta: miisiuc3m conoce a LaVerdad24ES son 24 horas de información, te invito a seguirlos.”

Capítulo 8

Conclusiones

Se finaliza el trabajo con este apartado de conclusiones tanto generales del proyecto, como personales en lo que ha significado su ejecución para mí.

También se recogen algunos trabajos que he detectado durante la ejecución del proyecto que si bien no son necesarios para el presente proyecto, pueden ser de utilidad y ayudar a ampliar el trabajo del proyecto.

8. Conclusiones

Sobre un contexto de uso masivo de las redes sociales en la sociedad actual y de un cada vez más creciente interés por la inteligencia emocional, el proyecto se inicia con un claro objetivo: Construir un modelo de conocimiento basado en análisis de sentimientos que sirva como base para el proceso de mejora en la labor docente de la asignatura Ingeniería de Sistemas de Información.

Para ello, los responsables de la asignatura habilitaron el perfil Twitter @miisi_uc3m y se encargaron de dar conocimiento del perfil y de los objetivos del mismo a los alumnos.

El proyecto se apoya en los datos intercambiados en el perfil para a partir de su análisis, obtener conocimiento. Un conocimiento basado en la interpretación de los sentimientos expresados en los *tweets*.

En un primer paso, se procede con la extracción de los *tweets* del perfil. El primer dilema que se plantea es ¿qué *tweets* son válidos para el estudio? Hay que tener en cuenta que además de tener acceso a *tweets* relativos a la asignatura, también hay *tweets* relacionados con el perfil que no se ajustan a las necesidades del estudio, por ejemplo aquellos que aparecen en el *home* del usuario por ser seguidor de cuentas generalistas como periódicos o publicaciones de tecnología. Los *tweets* objeto del análisis son las menciones y los *tweets* emitidos por el propio usuario.

Dentro de la labor de recolección de datos, un segundo paso consiste en la extracción de la información del perfil. La opción elegida aquí es el uso del API Twitter. Si bien, dado que la extracción necesaria para el proyecto es puntual, no siendo necesario realizar una extracción continua de información, elijo la utilización de una aplicación disponible en la propia web de Twitter y que permite explotar todo su API REST.

Una vez se dispone de todos los datos, tenemos que avanzar en la cadena de valor del proceso KDD mediante su conversión en información. Para ello en el proyecto se implementa una utilidad que transforma el *tweet* en formato JSON devuelto por Twitter, en lenguaje natural. El objetivo de esta conversión es disponer de los datos en un formato leíble tanto por humanos como por máquinas.

Llegados a este punto estamos en disposición de comenzar con la minería de sentimientos. En el proyecto, la minería de sentimientos se ha implementado apoyándose en el desarrollo de una ontología. En la ontología se conceptualiza el universo de *tweets* y se les dota de meta-propiedades y de relaciones mediante con cuyo análisis se logrará el objetivo final de convertir los *tweets* en conocimiento.

He seguido la siguiente metodología.

- Definición de términos.

  KnowledgeMANAGER viene pre-cargado con una gramática de lengua castellana, pero lógicamente no incluye el léxico propio de Twitter en general y el léxico particular de los *tweets* intercambiados en la asignatura, p.e. no incluye los *hashtags* específicos definidos por los profesores para expresar sentimientos.
  El primer paso por lo tanto ha consistido en abstraer esos términos específicos e incorporarlos al conjunto de términos disponibles en la herramienta.

- Taxonomía.

  En esta fase se ha procedido con la agrupación lógica de términos y relaciones entre dichas agrupaciones.

  Esta labor resulta imprescindible como un paso previo a la definición de la sintaxis donde se utilizarán estas agrupaciones y las relaciones entre las mismas, para definir la estructura de un *tweet*.

- Sintaxis/ Patrones

  En este paso completamos la formalización del *tweet* en una regla a la que llamamos patrón.
  El trabajo de formalización se realiza empezando por la elaboración de patrones sencillos. Mediante su combinación se elaboran patrones más completos y así sucesivamente hasta llegar a lo que hemos denominado patrones finales y patrones complejos donde se definen las reglas en las que se encuadran todos los *tweets*.

  Si bien todas las tareas realizadas hasta el momento así como el resto de tareas posteriores del proyecto resultan imprescindibles para la consecución del objetivo del estudio, me gustaría señalar esta tarea como la más importante del mismo y en consecuencia a la que más tiempo de desarrollo y análisis e implementación he dedicado.

  Todas las tareas realizadas hasta el momento de la definición de los patrones están orientadas a tener los tokens y agrupaciones necesarias para elaborar el patrón, el "*sujeto + verbo + predicado"* del *tweet*.
  Por otro lado, la definición de relaciones y meta-propiedades que haremos a continuación, se hace en base a los diferentes niveles de patrones definidos.

  Es por ello que una correcta definición de la estructura de patrones es vital para el éxito del estudio.

- Definición semántica / formalización

  Tarea consistente en asignar meta-propiedades a los patrones y definir relaciones entre elementos de los patrones del *tweet.* En definitiva, dotar al *tweet* de significado dentro del ámbito de los sentimientos.

Para finalizar con el proyecto, en base a la ontología creada, se produce el estudio de las relaciones y meta-propiedades indexando los *tweets*. El proceso de indexación instancia cada propiedad y relación en base a los datos de cata *tweet* de forma que podemos por ejemplo, conocer todos los usuarios que han publicado un *tweet*, el carácter de ese *tweet* (si es positivo, negativo, etc.) así como las relaciones de *tweet* de respuesta sobre otro *tweet.*

Concluyo el proyecto mostrando los resultados obtenidos tras la indexación con un modelo estadístico, modelo mediante el que se ***consigue el objetivo de dotar de CONOCIMIENTO a los DATOS de partida.***

8.1 Conclusiones personales

He querido incorporar también a este capítulo un conjunto de conclusiones personales sobre lo que ha significado la ejecución del proyecto para mí en el sentido de conocimientos adquiridos, dificultades encontradas y mi percepción global sobre el mismo.

En lo personal la ejecución de este proyecto fin de carrera me ha dado la oportunidad de conocer un campo en el que no había trabajado hasta el momento. Todas las tecnologías relativas a la minería de datos y extracción de conocimiento en base a ellos, es una rama muy de actualidad dado el gran volumen de datos que circula por internet y la importancia de incorporar conocimiento a la toma de decisión y definición de estrategias. En este sentido me siento satisfecho de haber podido aprovechar la oportunidad que me brinda la ejecución del proyecto para introducirme en esta área.

Desde el punto de vista tecnológico general, también me ha resultado de mucho interés profundizar en las técnicas de integración que se están ofreciendo hoy en día por los mayores portales de contenido en internet: tecnología REST o los métodos de autenticación y autorización basados en OAuth.

En cuanto a la mayor dificultad que me he encontrado en la ejecución del proyecto, ha sido la elaboración de la ontología. El hecho de no haber trabajado este campo con anterioridad ha hecho que mis primeros avances se produjesen de forma lenta, si bien es cierto que puedo decir que a día de hoy me siento bastante cómodo con el desarrollo del trabajo realizado. Día a día fui notando los progresos, en la fase de definición de los patrones tuve bastantes dificultades al principio, pero poco a poco me he ido sintiendo cómodo y motivado a la hora de elaborar nuevos patrones sobre todo al visualizar su resultado en la indexación de los *tweets*.

8.2 Futuras líneas de trabajo

Finalizo el capítulo y el proyecto indicando algunos puntos que considero de interés y que pueden dar continuidad al proyecto así como ampliarlo para nuevas ejecuciones del mismo sobre Twitter u otro tipo de interacciones alumno/asignatura.

El modelo y la metodología aplicada en el proyecto se han implementado sobre el universo de *tweets* objeto del estudio pero también teniendo en cuenta que el modelo se pueda aplicar a futuras experiencias de este tipo.

En este sentido, propongo a continuación algunos trabajos que si bien no han sido necesarios para la ejecución del presente proyecto, creo que pueden completarlo en estos tres aspectos.

- Dotarlo de herramientas que automaticen el proceso para su uso recurrente y ampliarlo a otras redes sociales.

  El proceso de extracción de datos del perfil de Twitter, si bien se ha realizado mediante la utilización de herramientas publicadas en la propia página de Twitter y que utilizan su API, ha sido realizado de forma manual. Tras la extracción de los datos con API Console, ha sido necesario copiarlos a mano y guardarlos en un fichero.

Si bien el trabajo no es costoso y es suficiente para la este proyecto, de cara a futuros procesos de extracción, podría ser interesante desarrollar una utilidad que mediante el uso del API REST, obtenga directamente los *tweets*.

Más aun pensando en que en este proyecto se ha utilizado Twitter como red social, pero el estudio podría extenderse a otras redes sociales o blogs propios de la asignatura. Por lo tanto la herramienta podría adaptarse para utilizar diferentes APIS y acceder a otras redes sociales como Facebook.

- Ampliar la ontología para ampliar el conjunto de *tweets* sobre los que pueda aplicarse el proceso de KDD.

  En la definición de los patrones hemos podido ver cómo se han definido en base a estructuras representativas de cada tipo de *tweet,* de forma que se han generalizado expresiones para facilitar que el análisis se ajuste a cualquier *tweet* que utilice expresiones habituales del lenguaje.

  Aun así, la ontología podría completarse con un entrenamiento de los patrones en el sentido de extraer más volumen de información e indexarlo con los patrones existentes. De esta forma podrían encontrarse nuevas expresiones que puedan enriquecer la ontología.

- Implementar mecanismos de exportación de la indexación.

  Todo el mecanismo de indexación de los *tweets* se realiza de una forma bastante sencilla, pero se echa en falta la posibilidad de exportar los resultados.

  Para el desarrollo del punto 7.1 Resultados del Análisis, se ha realizado un proceso de traslado manual de la información a Microsoft Excel para a partir de uso de tablas dinámicas y gráficos enlazados a las tablas, poder dotar a las conclusiones de un análisis agrupado y gráfico de los resultados.

  Mi propuesta en este sentido sería dotar a knowledgeMANAGER de mecanismos de exportación de los resultados de la indexación a diferentes formatos como puede ser texto plano o XML.

Capítulo 9

Referencias

A continuación se muestra un índice con todas las referencias nombradas en el documento.

9. Referencias

[1] UNESCO, "Hacia las sociedades del conocimiento," 2005. [Online]. Available: http://unesdoc.unesco.org/images/0014/001419/141908s.pdf. [Accessed Marzo 2015].

[2] J. C. R. Licklider, «IRE Transactions on Human Factors in Electronics,» MIT, [En línea]. Available: http://groups.csail.mit.edu/medg/people/psz/Licklider.html. [Último acceso: Marzo 2015].

[3] «ISC Domain Survey,» Internet Systems Consortium, 2015. [En línea]. Available: https://www.isc.org/network/survey/. [Último acceso: Marzo 2015].

[4] ITU, «WTID 2014 Statistics,» [En línea]. Available: http://www.itu.int/en/ITU-D/Statistics/Documents/statistics/2014/stat_page_all_charts_2014.xls. [Último acceso: Marzo 2015].

[5] W3C, «Tim Berners-Lee Biography,» [En línea]. Available: http://www.w3.org/People/Berners-Lee/. [Último acceso: Marzo 2015].

[6] BLOGGER, «https://www.blogger.com,» BLOGGER, [En línea]. Available: https://www.blogger.com/about. [Último acceso: Junio 2015].

[7] M. Mullenweg, «WordPress Now Available,» https://wordpress.org, [En línea]. Available: https://wordpress.org/news/2003/05/wordpress-now-available/. [Último acceso: Marzo 2015].

[8] ONTSI, «http://www.ontsi.red.es/ontsi/,» Ministerio de Industria Energía y Turismo, Diciembre 2011. [En línea]. Available: http://www.ontsi.red.es/ontsi/es/estudios-informes/estudio-sobre-el-conocimiento-y-uso-de-las-redes-sociales-en-españ. [Último acceso: Junio 2015].

[9] «http://www.classmates.com,» Classmates, [En línea]. Available: http://www.classmates.com/about/. [Último acceso: Abril 2015].

[10] Alexa, «http://www.alexa.com,» Amazon - Alexa, [En línea]. Available: http://www.alexa.com/topsites. [Último acceso: Mayo 2015].

[11] Twitter, «http://twitter.com,» Twitter, [En línea]. Available: https://about.twitter.com/company. [Último acceso: Mayo 2015].

[12] Eric Schmidt, «http://readwrite.com,» [En línea]. Available: http://readwrite.com/2011/02/07/are-we-really-creating-as-much. [Último acceso: Marzo 2015].

[13] E. Punset, «Nadie nos ha enseñado a sonreír,» [En línea]. Available: http://www.eduardpunset.es/tag/inteligencia-emocional#sthash.g9VSVrQs.dpuf. [Último acceso: Agosto 2015].

[14] M. N. Z. C. Daniel Goleman, «Slideshare - Inteligencia Emocional,» [En línea]. Available: http://es.slideshare.net/sistematizacion/inteligencia-emocional-1144462. [Último acceso: Julio 2015].

[15] C. C. Montiel, «Extracción de información social desde Twitter,» Septiembre 2012. [En línea]. Available: http://e-archivo.uc3m.es/bitstream/handle/10016/16784/memoriaTFG_Cristian_Caballero_Montiel.pdf?sequence=1. [Último acceso: Agosto 2015].

[16] Twitter, «The Streaming APIs,» [En línea]. Available: https://dev.twitter.com/streaming/overview. [Último acceso: Mayo 2015].

[17] P. R. Fernández, «Aplicación de las nuevas tecnologías web para clasificación de contenidos en redes sociales,» [En línea]. Available: http://e-archivo.uc3m.es/bitstream/handle/10016/16669/MemoriaPFC_Pablo_Rodriguez_Fernandez.pdf?sequence=1. [Último acceso: Julio 2015].

[18] Wikipedia, «Pooling,» [En línea]. Available: http://es.wikipedia.org/wiki/Polling. [Último acceso: Marzo 2015].

[19] S. koiral, «Implementing 5 important principles of REST using WCF Services,» Febrero 2014. [En línea]. Available: http://www.codeproject.com/Articles/283550/Implementing-important-principles-of-REST-using. [Último acceso: Mayo 2015].

[20] JSON, «JSON JavaScript Object Notation,» [En línea]. Available: JavaScript Object Notation. [Último acceso: Junio 2015].

[21] Twitter, «The Search API,» [En línea]. Available: https://dev.twitter.com/rest/public/search. [Último acceso: Mayo 2015].

[22] Twitter, «REST APIs,» [En línea]. Available: https://dev.twitter.com/rest/public. [Último acceso: Mayo 2015].

[23] J. A. F. Autores: Norberto Fernández García, «OCW - REpresentational State Tranfer (REST),» [En línea]. Available: http://ocw.uc3m.es/ingenieria-telematica/tecnologias-de-distribucion-de-contenidos/transparencias_tdc/rest/at_download/file. [Último acceso: Mayo 2015].

[24] S. Tilkov, «A Brief Introduction to REST,» Diciembre 2007. [En línea]. Available: http://www.infoq.com/articles/rest-introduction. [Último acceso: Mayo 2015].

[25] Internet Engineering Task Force (IETF), «Hypertext Transfer Protocol -- HTTP/1.1,» 1999. [En línea]. Available: http://tools.ietf.org/html/rfc2616. [Último acceso: Mayo 2015].

[26] Internet Engineering Task Force (IETF), «Uniform Resource Identifier (URI): Generic Syntax,» Enero 2005. [En línea]. Available: http://tools.ietf.org/html/rfc3986. [Último acceso: Mayo 2015].

[27] Internet Engineering Task Force (IETF), «The OAuth 2.0 Authorization Framework,» Octubre 2012. [En línea]. Available: http://tools.ietf.org/html/rfc6749. [Último acceso: Abril 2015].

[28] Hueniverse, «OAuth,» [En línea]. Available: http://hueniverse.com/oauth/. [Último acceso: Abril 2015].

[29] TICbeat, «Facebook OAuth 2.0,» Abril 2010. [En línea]. Available: http://www.ticbeat.com/analisis/porque-facebook-cambia-oauth-estandar-abierto-autentificacion-twitter/. [Último acceso: Mayo 2015].

[30] Error500, «OAuth adoptado por Gooble,» [En línea]. Available: http://www.error500.net/oauth-adoptado-por-google/. [Último acceso: Abril 2015].

[31] Apigee, «APIgee Twitter Console,» [En línea]. Available: https://dev.twitter.com/rest/tools/console. [Último acceso: Abril 2015].

[32] Ignacio Silla Ruíz de Cenzano, Cristina Sanz Sánchez, «Identidad en la Red y portabilidad de datos personales,» 2010. [En línea]. Available: http://es.slideshare.net/sgua/identidad-en-la-red-y-portabilidad-de-datos-personales. [Último acceso: Junio 2015].

[33] Hueniverse, «OAuth Terminology,» [En línea]. Available: http://hueniverse.com/oauth/guide/terminology/. [Último acceso: Abril 2015].

[34] Hueniverse, «Protocol Workflow,» [En línea]. Available: http://hueniverse.com/oauth/guide/workflow/. [Último acceso: Abril 2015].

[35] Twitter, «twitter/snowflake,» [En línea]. Available: https://github.com/twitter/snowflake. [Último acceso: Junio 2015].

[36] Twitter, «REST APIs,» Twitter, [En línea]. Available: https://dev.twitter.com/rest/public. [Último acceso: Mayo 2015].

[37] Twitter, «Twitter - Terms of Use,» Twitter, [En línea]. Available: https://dev.twitter.com/overview/terms. [Último acceso: Abril].

[38] Twitter, «Tweets API Overview,» [En línea]. Available: https://dev.twitter.com/overview/api/tweets. [Último acceso: Junio 2015].

[39] CIO, «10 Free Essential Twitter Tools for Power Users,» [En línea]. Available: http://www.cio.com/article/2390734/consumer-technology/10-free-essential-twitter-tools-for-power-users.html. [Último acceso: Agosto 2015].

[40] t. R. company, «the REUSE company knowledge MANAGER,» [En línea]. Available: http://www.reusecompany.com/knowledgemanager33. [Último acceso: Junio 2015].

[41] ic#code, «ic#code,» [En línea]. Available: http://www.icsharpcode.net/OpenSource/SD/. [Último acceso: Julio 2015].

[42] Mono Project, «Mono Project,» [En línea]. Available: http://www.mono-project.com. [Último acceso: Julio 2015].

[43] KNOWLEDGE REUSE GROUP, «Knowledge Reuse Group,» [En línea]. Available: http://www.kr.inf.uc3m.es/index.php. [Último acceso: Agosto 2015].

[44] Twitter, «Users API Overview,» [En línea]. Available: https://dev.twitter.com/overview/api/users. [Último acceso: Mayo 2015].

[45] The REUSE company, «knowledgeMANAGER User Guide (Spanish),» [En línea]. Available: http://www.reusecompany.com/technical-documentation-km/knowledgeMANAGER---Technical-Documentation/0---knowledgeMANAGER-User-Guide-(Spanish)/. [Último acceso: Agosto 2015].

[46] Real Academia Española, «Diccionario de la Real Academia Española,» [En línea]. Available: http://lema.rae.es/drae. [Último acceso: Septiembre 2015].

[47] T. Berners-Lee, «WWW User guide,» CERN, [En línea]. Available: http://www.w3.org/History/19921103-hypertext/hypertext/WWW/TheProject.html. [Último acceso: Septiembre 2015].